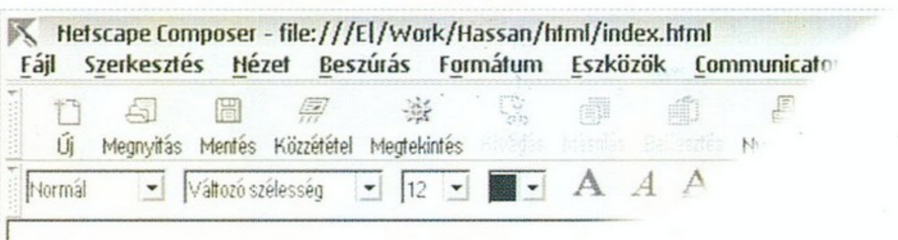
Magyar számítástechnikai folyóirat CD-melléklettel

Ára: 648 Ft

XVII. évfolyam 2. szám

1999. február

ÚJ ALAPLAP



A hónap témája:
MIT HOZ A HONLAP?



Microsoft és a szabad forráskód
Nyílt tér

Calmira: továbbra is 16 biten
Szerszámosláda

Jönnek a Windows scriptvírusok
Vírusőrjárat

Tivoli, az univerzális menedzser
Hálózat

Digitális hátfalak
Fogódzó



99002

9 771217 759005



Vegyen Ön is lendületet! Akár már havi 2.000 forintért!

ElenderMan begyújtotta az ISDN-rakétákat, hogy legyőzzön Időt és Teret. Fizessen elő az Elender ISDN-szolgáltatására, és Ön is gyorsabban, hatékonyabban internetezhet, mint telefonvonalon – ugyanannyiért. Mert ezentúl csak az Interneten töltött idő számít. Azaz már havi 2.000 forintért is internetezhet ISDN-en, havi 10.000 forintért pedig korlátlan hozzáférés jár + akár **60.000** forintnyi ajándék is! Hiszen a jó teljesítményen is lehet ugrásszerűen javítani.

ISDN Internet akció!

elender internet

Semmi sem lehetetlen.

INFORMÁCIÓKÉRÉS: 10 ▲

Elender Informatikai Rt. Tel.: 465-7800; Fax: 465-7899; 1134 Budapest, Váci út 37. http://www.elender.hu

ÚJ ALAPLAP

A Mikroszámítógép Magazin és az Alaplap hagyományait folytató magyar számítástechnikai folyóirat

Megjelenik havonta, CD-melléklettel

Főszerkesztő:
Faklen Pál

Szerkesztő:
Jakab Ágnes

A szerkesztőbizottság tagjai:
Adorjáni Gábor, Ambrózy Gábor, Aszalós László, Bánó György, Feleki Zoltán, Galántai Zoltán, Herczeg József, Horlai János, Kádár Zsolt, Kovács Attila, Mákos András, Pogány Csaba, Protzner György, Simay Endre István, Szappanos Gábor, Szondi Egon János, Vargha Dénes, Vékony Tamás

Szerkesztőség és kiadó:
1539 Budapest, Pf. 571
VI., Dózsa György út 84/b
Telefon: 322-4417, 322-5238
Fax: 351-8015
E-mail: alaplap@mail.datanet.hu

Weblap: http://www.alaplap.hu

Felelős kiadó:
Faklen Pál

Terjesztés:
Megyes Zsuzsanna

Hirdetésszervezés:
Árvai Katalin, Bogácsi Mária, Galyasi Hedvig, Pap Katalin

Külföldi hirdetések:
PubliCity
Reklám- és Médiaügynökség
1537 Budapest I., Márvány u. 17.
Telefon: 356-1182 Fax: 375-3539

A kiadó a hirdetések tartalmáért és a nyomdakészen kapott hirdetések formájáért (és helyesírásáért) nem vállal felelősséget

Példányszámadatok hitelesítése:
Magyar Terjesztésellenőrző Szövetség
MATESZ
Ez a szám 9000 példányban jelent meg

Nyomtatás:
Zalai Nyomda Rt, Zalaegerszeg
Felelős vezető:
Czirkl György vezérigazgató

Terjeszti:
A Magyar Posta Rt, a Nemzeti Hírlapkereskedelmi Rt, a Hírker Rt, a Kiadói Lapterjesztő Kft és számos számítástechnikai szaküzlet

Előfizethető a kiadónál:
Új Alaplap Kiadói Kft,
1539 Budapest, Pf. 571
Bankszámlaszám:
OTP 11706016-20788599

A lap példányonkénti ára: 648 Ft
Évi előfizetési díj: 6480 Ft

Külföldi előfizetés díja:
6480 Ft + postázási költség

HU ISSN 1217-7598

TARTALOM

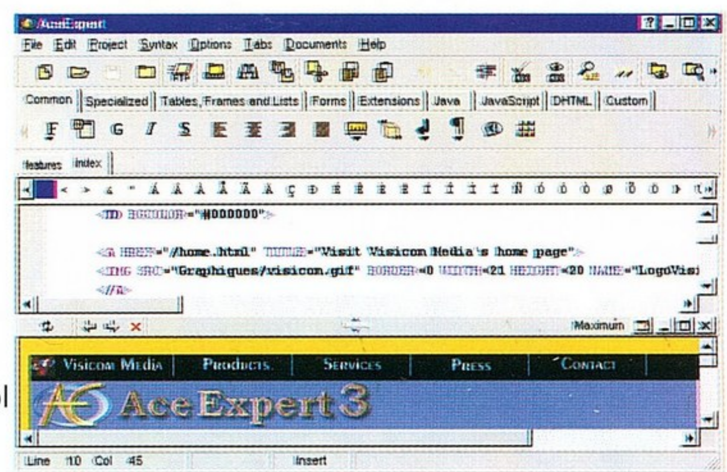
A CD-MELLÉKLETEN

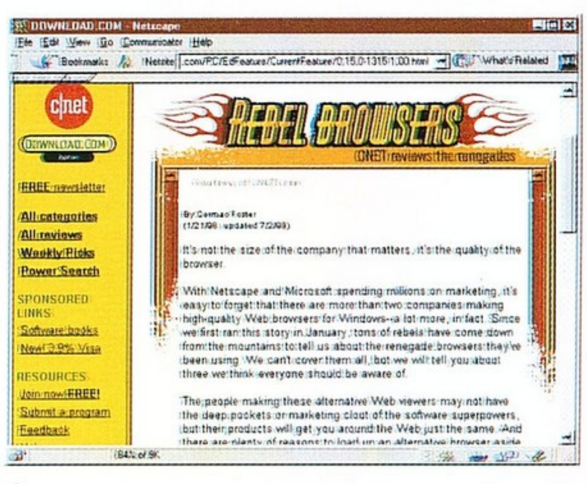
1999. FEBRUÁR

FOKUSZ	*Fókuszban a webszerkesztés*
DELPHI	Delphi segédeszközök HTML-hez, Javához, Internethez
DOC	Webbel kapcsolatos dokumentációk (HTML, XML, CSS stb.)
EDITOR	HTML-editorok, site-editorok, HTML segédeszközök
1-4-ALL	1-4-All v2.1 (Win32)
AARDVARK	Aardvark Pro 3.01 (Win32)
ACEEXPER	AceExpert v3
AGILE	Agile
ANANSI	Anansi (Win16 és Win32)
AOLPRESS	AOLPress 2.0 (Win32)
ARACHNO	Arachnophilia
CLARIS	Claris Home Page 3.0
COFFECUP	CoffeCup v5.5
COOLCAT	CoolCat 4.0
DIPRO250	DiDapro V2 (Win16 és Win32)
DREAMWAE	Macromedia Dreamweaver
DRUMBEAT	Elemental Drumbeat (Win32)
HIPPIE	Hippie 5.0 (Win16) és Hippie 98
HOMESITE	Homesite 1.2 freeware, Homesite 4.0 próbaverzió
HOTDOG	HotDog (Win16 és Win32)
HOTMETAL	SoftQuad HotMetal Pro 5.0 (Win32)
HTMLED32	HTMLed32
HTMLEDPRO	HTMLed Pro 2.5
HTMLPAD	HTMLpad 2.6
HTMLWRITE	HTMLWrite
LOTUSFS2	FastSite 2.0
NETOBJEC	Netobjects Fusion 4.0 site-editor
PAGEMILL3	Adobe PageMill 3.0
SITEWRIT	SiteWriter Plus
SYMANTEC	Symantec Visual Page 1.1
SYNTAX	Syntax v1.0
TARANT	Tarantula v1.98
WEBEDIT	WebEdit (Win16 és Win32)
WEBEXPR	WebExpress 3.0 (Win16)
SEGED	Segédletek HTML-szerkesztéshez
HATTER	Képek weblapok hátteréhez (textúrák)
MICROSFT	
DINHTML	Microsoft Dynamic HTML Editing Component SDK
IE5	Internet Explorer 5.0 béta-változat
SCR50N	VBScript és JScript futtatóeszköz az IE 5-öshöz
REBBROWS	Rebellis böngészők (Minden, ami nem Netscape, és nem Explorer...)
AMAYA	Amaya 1.3b (Win32)
ARACHNE	Arachne DOS-os böngésző
HOTJAVA	HotJava
LYNX	Lynx 2.8.1 karakteres böngésző forráskóddal
MOSAIC	Mosaic böngésző
OPERA	Opera (16 és 32 bites)
SOFTERM	Softerm Plus+ (16 bites böngésző)
SHAKESP	Shakespeare dráma XML formátumban
LAPFORG	*Lapraforgó*
CALMIRA	Calmira, Windows 95-ös felület 16 biten!
EXCEL	A sorozat zárórészéhez tartozó állományok
FOTO	A digitális hátfalakról szóló cikk illusztrációi
GNU	A GNU for DOS sorozathoz tartozó anyagok
HALLOWEE	Microsoft dokumentum a szabad forráskódokról
MICROSFT	Microsoft Script Control
SZERSZAM	*Szerszámosláda*
EURO	Euro kiegészítés Windows 95-höz és NT-hez
FESZER	Fészer, a gyakran szükséges segédprogramokkal
VIRUS	Vírusirtók
DOS	DOS-os alkalmazások
WIN3.x	16 bites windowsos alkalmazások
WIN9.x	32 bites windowsos alkalmazások
LINUX	Linuxos alkalmazások
OS2	OS/2-es alkalmazások
VENDEG	*Vendégoldal*
ADOBE	Adobe programfejlesztési segédlet (SDK)
DBTOOLS	Segédeszközök dBase és Paradox adatbázisokhoz
KKERESO	A Webhu „Kincsereső” offline változata
WORDPER	Corel WordPerfect 8 Linuxra, personal edition
WORDSER	Corel WordPerfect 8 Linuxra, server edition
JATEKVAR	*Játékvár*









Az amatőrizmus alkonya

Webszerkesztés és HTML

Egyelőre a (World Wide) Web és a HTML (Hypertext Markup Language, azaz hipertextes jelölőnyelv) közé még akkor is majdhogynem egyenlőségjelet tehetünk, ha az utóbbi időben egyre többen gondolják úgy, hogy végül mégsem ez lesz az egyedül üdvözítő megoldás. Jelenleg természetesen leginkább a szakértők vallják ezt, vagyis azok, akik nem érik be azzal, hogy egy óra alatt összedobjanak egy honlapot magukról meg a kedvenc kutyájukról.

A HTML nagy előnye, hogy lényeges elemeit az ember még unalomból is meg tudja tanulni egy esős délután, illetve aki még erre is lusta, az néhány weboldalt valamelyik HTML-editorral jól-rosszul, de általában nem teljesen használhatatlan formában képes megírni (a tartalomról most tapintatból ne beszéljünk). Mivel azonban semmi sincs ingyen, valahol ezért az egyszerűségért és a programnyelvekhez képest könnyű kezelhetőségért is fizetni kell: az egyszerűbb megoldásokkal beérő felhasználó elégedett lehet a HTML által felkínált lehetőségekkel, de aki bonyolultabb feladatot akar megoldani vele, annak ez már nehézkes, bizonyos esetekben pedig egyenesen alkalmatlan eszköz.

Nem tehetjük meg például, hogy saját elemeket definiáljunk benne, hanem kizárólag a rendelkezésünkre álló készletből építkezhetünk, az így létrejövő struktúra tehát meglehetősen merev lesz, az egyéni elképzelések megvalósítására nem sok teret enged. Márpedig az „egyéni igények" jelen esetben a világ legnagyobb, immár a Web felé is terjeszkedő cégeit jelentik. Sejthető tehát, hogy a következő egy-két évben valóban alapvető változások fognak bekövetkezni.

1993 júniusában Matthew Gray „World-Wide-Web Wanderer" nevezetű szoftvere az egész Internetet végigkeresve is alig 100 home page-et tudott felfedezni, 1994 márciusában viszont már 1100-at, és mindössze két hónappal később Brian Pinkerton ma is működő „WebCrawler"-je több mint 3000-et. Azután, amikor a hagyományos média is felfedezte magának a Webet, és a CNN, a Wall Street Journal, az Economist és a Fortune is rátelepült, kezdetét vette az az első korszak, amelyet legegyszerűbben a HTML dominanciájával és a szó jó értelmében vett amatőrizmussal jellemezhetünk.

Még 1996-ban is (sőt, olykor 1997-ben is) elképzelhető volt, hogy egy professzionális igényekkel fellépő megrendelő a weblapokat mintegy hobbiból írogatóhoz forduljon. 1999 elejére azonban mindenki számára nyilvánvalóvá vált, hogy egy igényesebb honlap megírása egyrészt ugyanúgy külön szakma, mint egy igényesebb folyóirat elkészítése, másrészt ugyanúgy csapatmunka is, tehát egy ember gyakorlatilag nem elég hozzá. Legalább ketten kellenek hozzá: az egyik (nevezzük szerkesztőnek) a tartalommal, a html összeállításával foglalkozik, a másik (nevezzük grafikusnak) az alkalmazott grafika új ága képviselőjeként a képeket, az animációkat, az ikonokat, a háttért stb. készíti el. Minél bonyolultabb, és a kor kihívásainak jobban megfelelő weblapról van szó, annál inkább követelmény továbbá, hogy legalább az egyikük — vagy egy harmadik közreműködő (nevezzük programozónak) — értsen a Javához, a CGI-hez, a Perlhöz és hasonlókhoz.

Mindez természetesen nem azt jelenti, hogy az „egyszemélyes amatőröknek" fel kellene hagyniuk a weblapok írásával, hanem egyszerűen arról van szó, hogy a feladatok összetettsége és növekvő mennyisége bizonyos munkamegosztást tesz szükségessé a komolyabb weblapok rendszeres működtetése során. Akinek kedve van, az továbbra is nyugodtan felteheti saját weblapjára a vasárnapi kirándulásról készített amatőrfelvételeit, és ebben senki nem fogja



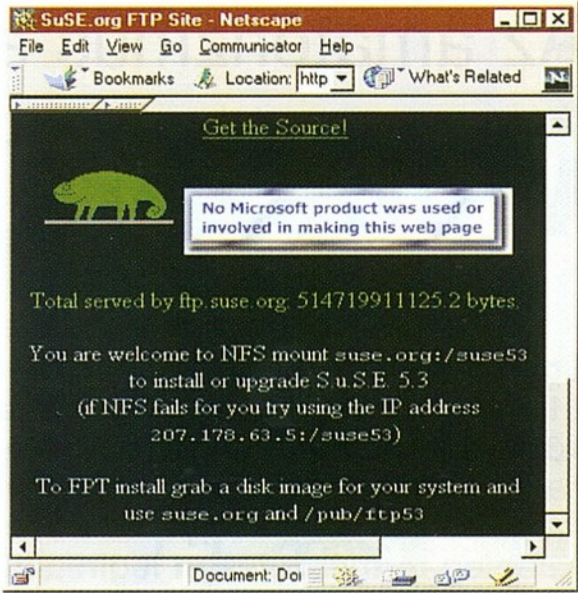
megakadályozni. Másfelől viszont az is biztos, hogy tovább nő a különbség az általa készített és a profi weblaptervezők által kreált oldalak között.

Talán még jobb példa a könyvkiadás. Valóban el lehet készíteni egy könyv oldalainak eredetijét WinWord x.x-ben is, de ránézésre azonnal felismerhető, hogy nem a könyv professzionális kiadványszerkesztői fázisáról van szó, hanem legfeljebb egy jól összeállított nyersanyagról. Ugyancsak van különbség a WinWordben megírt és HTML-re konvertált anyagok és a „rendes" HTML-szerkesztővel megírt weblapok között. És a szoftver önmagában nem is elég: én például a világ legjobb programjával sem nagyon tudnék megfelelően betördelni egy könyvet (az újságokról nem is szólva). A Web azonban kísérletezésre csábít: egy házilag összeütött honlapnak ugyanis gyakorlatilag nincsenek költségei. Ha ugyanebből kereskedelmi forgalomba kerülő nyomtatott terméket készítenénk, nagy valószínűséggel jókorát buknánk vele anyagilag.

A Web egyik sajátossága feltehetően a jövőben is az lesz, hogy a papíralapú publikációkhoz viszonyítva elképesztő mértékben és igen nagy százalékos arányban lesznek jelen az amatőrök és a dilettánsok, akik azonban többnyire hozzáértésüket jóval meghaladó eszközrendszerrel fognak dolgozni. Ennek persze vannak „hagyományai", már jó ideje kénytelenek vagyunk naponta olvasgatni a legjobb kiadványszerkesztőkkel készített ízléstelen és csiricsáré körleveleket, és bámulni a legkorszerűbb grafikai programokkal készített gyenge hirdetéseket meg illusztrációkat.

A skála másik végpontján értelemszerűen a profi webkreátorok helyezkednek majd el, olyanok, akik egyénileg vagy csoportmunkában képesek a rendelkezésükre álló technológia lehetőségeit célirányosan kihasználni, ízlésesen alkalmazni.

Ami persze még mindig csak a dolog egyik fele. A másik az, hogy remélhetőleg hamarosan túljutunk a statikus HTML-re és az amatőrökre épülő, fent emlegetett első webkorszakon, és kezdetét veszi a második periódus, amelyben kifinomultabb eszközük állnak majd redelkezésünkre, és nem kell a



HTML korlátaival és megkötöttségeivel küszködnünk.

A HTML nyilván nem fog egyik pillanatról a másikra eltűnni, és az is elképzelhető, hogy miközben a weblapírás a profik kezében egyre inkább a programozáshoz közelít, a HTML közvetítőnyelvként még nagyon sokáig fennmarad. És ez akkor sem olyan nagy baj, ha használói közül korántsem lesz belőle mindenkinek legalább középfokú nyelvvizsgája.

Galántai Zoltán

Tim Berners-Lee és a Web forrásvidéke

Honnan jött, hová megy?

Bár egy évtizednél is rövidebb ideje létezik, az Internet egyik alkalmazása már olyannyira maga alá gyűrte lehetséges vetélytársait, hogy sokak számára a World Wide Web maga „A hálózat". Érdemes hát foglalkoznunk egyfelől azzal a kutatóval, akitől az egész származik, másfelől pedig innen elindulva a Web jövőjével is.

„A jövő webtechnológiájának az elektronikus kereskedelemre elképesztő hatása lehet. Képzeljük el például, hogy azt mondjuk: szükségünk van néhány 100 wattos, energiatakarékos izzóra, keressük meg tehát az ilyet árusító cégeket! A webprogram pedig (akár böngésző is lehet) nekiáll végignézni az esetleg teljesen különböző formátumú katalógusokat, kiválasztja az általunk megadott specifikációval rendelkező termékeket, létrehoz belőlük egy adatbázist, kiválasztja a leginkább megfelelőeket, és akár meg is rendeli azokat nekünk. Hiba lenne azt hinni, hogy akár az Internet, akár a Web már készen van. Még csak az elején tartunk, és rengeteg mindent kell kifejleszteni..." (Tim Berners-Lee, 1998. június 8., Los Angeles Times)

A világháló iránt érdeklődők körében közismert tény, hogy az Internet 1969-ben, az Egyesült Államokban jött létre, amikor katonai megfontolások alapján létrehoztak egy decentralizált, négy csomópontú számítógépes hálózatot. Ehhez kezdetben főleg egyetemek és kutatóintézetek csatlakoztak, így többnyire a műszaki-tudományos értelmiség szűk rétege használta azt. Később kiderült, hogy sokkal nagyobb lehetőségek rejlenek benne, ha a kezelőfelület barátságosabbá tehető. Ennek atyja Tim Berners-Lee, a genfi CERN részecskefizikai kutatóintézet egyik munkatársa volt, aki új módszert javasolt az intézeti hálózat adatainak kényelmes és hatékony használatára, ami az általa írott protokollra és a hipertextre épült. A protokollt, illetve a szoftvert, amely Enquire (Tudakozódj) néven a Norsk Data gépeken futott, már 1980-ban megírta, de akkoriban még nem tette publikussá. 1989-ben azután visszatért elgondolásához, és amikor 1990-ben másodszor is az igazgatósághoz fordult, a projekt végre szabad utat kapott.

Az angol matematikus családból származó Tim Berners-Lee 1976-ban szerzett diplomát az angliai Oxford Egyetemen. Egyetemi évei alatt elkészítette első számítógépét — ez egy M6800-as processzorból, egy régi tévéből és néhány egyéb elektronikai alkatrészből tevődött össze.

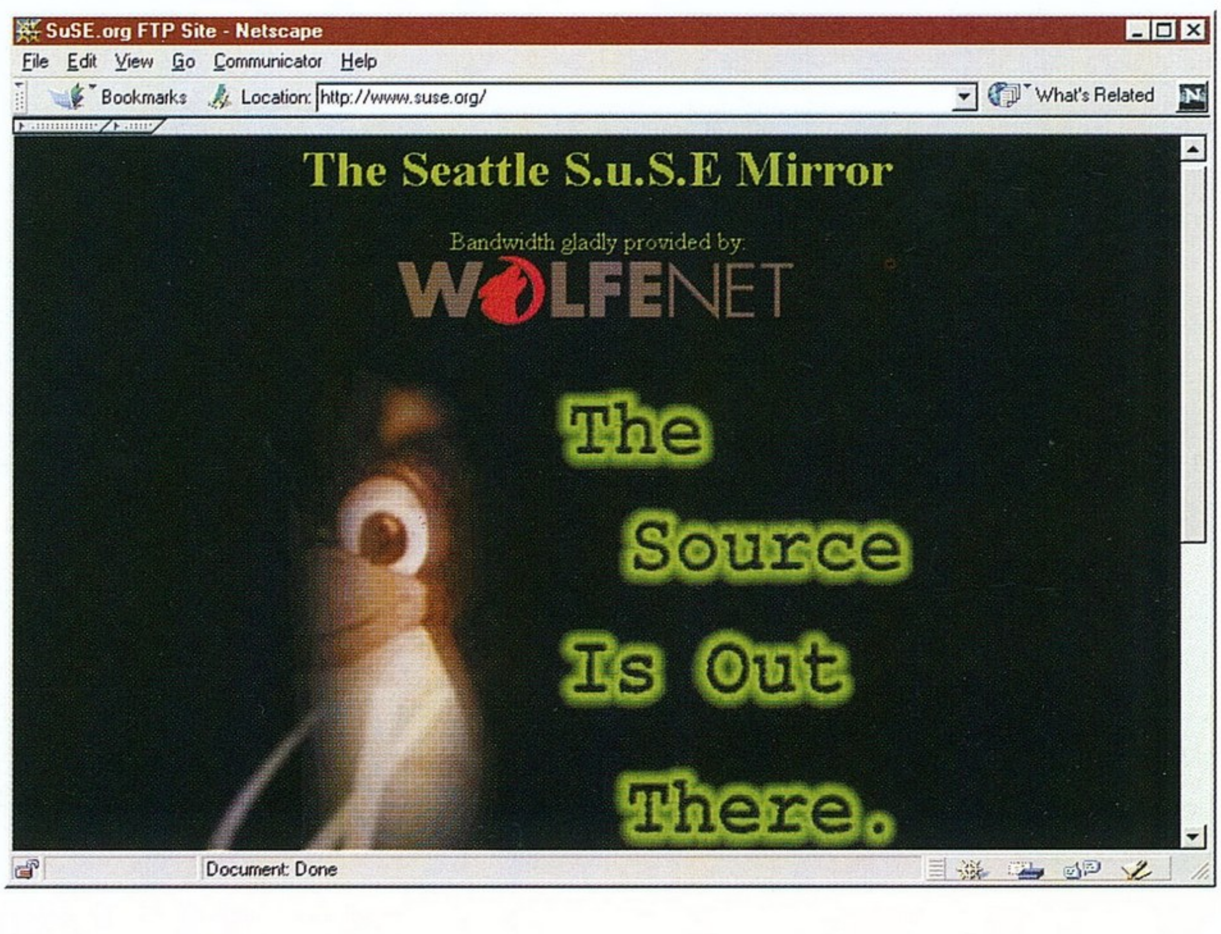
Tanulmányai befejeztével egy nagy brit telekommunikációs cégnél kapott állást, majd egy év múlva munkahelyet váltott, és intelligens nyomtatók számára írt különböző szoftvereket, illetve készített egy multitasking operációs rendszert is. Genfben a CERN-nél (az Európai Részecskefizikai Kutatóintézetnél) független tanácsadóként eltöltött másfél évet, ebből hat hónapot programfejlesztőként. Ekkoriban írta azt az Enquire nevű programot, ami soha nem vált ugyan széles körben ismertté, ám később mégis alapvető szerepet játszott a Web koncepciójának kifejlesztésében.

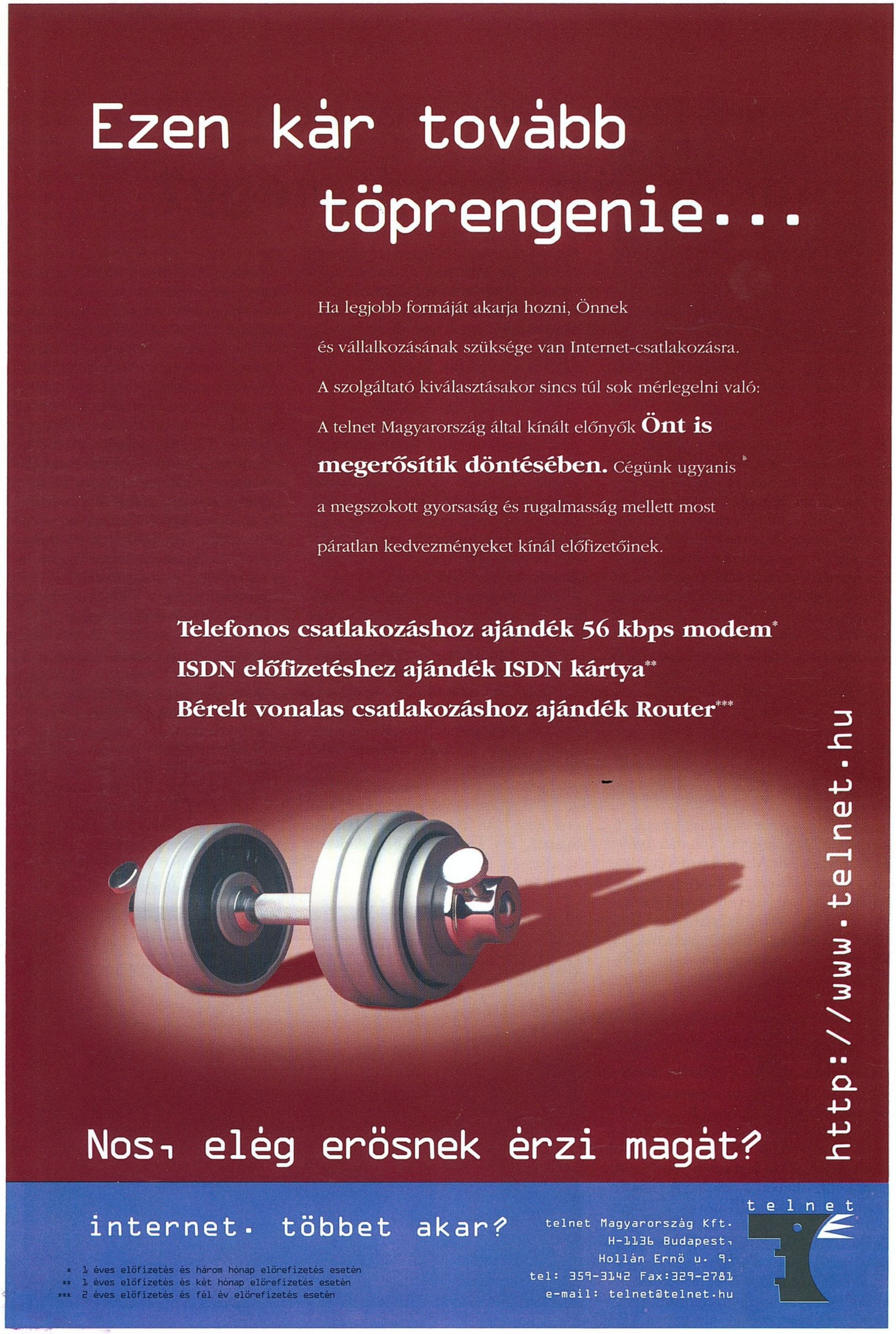
Berners-Lee 1981-től 1984-ig másik cégnél dolgozott tervezőként, majd tudományos munkatársként ismét állást vállalt a CERN-nél. 1989-ben állt elő azzal az átfogó hipertextes elképzeléssel, amelyből végül a Web megszületett.

Főnöke engedélyével nekivágott a technológia kidolgozásának. NeXTStep platformra kezdett fejleszteni egy hipertext GUI böngészőt és editort, a programnak pedig a World Wide Web nevet adta, mert már elege volt a görög mitológiából kölcsönzött nevekből. Munkatársa, Robert Cailliau segítségével mintegy négy hónap alatt készült el az első, ún. line mode (sormódú) browser és editor, a következő év (1990) májusában pedig a CERN-nél beindították az első WWW hostgépet.

Közben Berners-Lee (illetve ekkor már nemcsak ő, hanem egy egész hipertext csoport) kidolgozta a dokumentumok egységes formában való kezelését lehetővé tevő HTML-t (Hypertext Markup Language), és innen kezdve nem volt megállás. A nyelv hatékonyságának javításával böngészőik egyre jobbak lettek, az Interneten sorra jelentek meg a HTTP szerverek.

Ekkor más mások is nekiálltak böngészőket írni, és azok közül az NCSA (National Center for Supercomputing Applications) Mosaic nevű grafikus





INFORMÁCIÓKÉRÉS: 23 ▲

böngészője bizonyult a legfontosabbnak, márcsak azért is, mert ez volt az első grafikus point-and-click browser, a látvány pedig igencsak számított. Az is sokat nyomott a latban, hogy a New York Times 1993 decemberében egy többhasábos cikket jelentetett meg a Webről és a Mosaicról. Jelentős részben a Mosaic böngészőnek volt köszönhető, hogy a Web grafikus arculatúvá és hirtelen igen ismertté és népszerűvé vált. Berners-Lee és csapata számára pedig a WWW térhódításával párhuzamosan egyre nyilvánvalóbb lett, hogy a spontán folyamatokat valamilyen mederbe kell terelni, és ehhez bizony szervezőmunka kell.

1991 és 1993 között folytatódott a Web kiépítése a felhasználók visszajelzései alapján. Az egységes címzés (URL, Uniform Resource Locator), az adatátvitel (HTTP, Hypertext Transfer Protocol) és a HTML szabványait, illetve előzetes terveit finomították, különböző konferenciákon megvitatták. 1994-ben azután Berners-Lee csatlakozott az MIT (Massachusetts Institute of Technology) számítógép-tudományi laboratóriumához, ahol a Web kézben tartása érdekében megalakították a nonprofit szervezetként működő W3C-t (World Wide Web Consortium), amelynek fő célja akkoriban az adatbiztonság megteremtése volt. Manapság viszont inkább az, hogy összetartsa a Webet mint egységes rendszert, és megfelelő standardokat dolgozzon ki (a részletekről lásd a Konzorciummal foglalkozó cikket a 8. oldalon).

Maga Berners-Lee mondta, hogy a W3C legtöbbször tüzet olt, de olykor facsemetéket is nevel.

Berners-Lee-vel mostanában igen sok interjú készül. Ezekben gyakran beszél a Webet a jövőben forradalmasító módszerekről és eljárásokról. Ilyen lenne az XML (lásd az XML-ről szóló cikket a 20. oldalon), a többnyelvű publikálás, a valóban nemzetközivé válás, az online hírszolgáltatás, az elektronikus kereskedelem megújulása, a téma szerinti adatszűrés (a szülők és iskolák kívánsága szerint) stb. Az új elképzelésekben tulajdonképpen arról a javított architektúráról van szó, amely szerencsés esetben a WWW második forradalmához vezethet.

Eddig ugyanis a HTML többnyire az ember és ember közötti kommunikációt támogatta, az új struktúra, az RDF (Resource Description Format) viszont több lehetőséget nyújt a gépek közötti kommunikációra. Az információk a számítógép számára eddig legfeljebb olvashatóak voltak, de nem voltak megérthetőek, az RDF éppen ezen fog változtatni, mert olyan környezetet teremt majd a szoftverágenseknek, hogy azok képesek lesznek az előre definiált szabályok alapján online feladatokat megoldani. Ilyen volna például az intelligens, összehasonlításokon alapuló vásárlás, amikor egy meghatározott feltételrendszer megadása után a program a Weben maga dönti el, hogy mi az optimális választás.

A számítógép sok olyan tevékenységet végezhet el, amely fellendíti az elektronikus kereskedelemet, és nem csak az üzleti szféra számíthat fejlődésre. Az persze mindenképpen. A marketinghez már ma is egyre inkább hozzátartozik az Interneten (Weben) való jelenlét, és ismerve a pénz hatását, ez jelentősen befolyásolhatja az ottani technológiák fejlődési irányát.

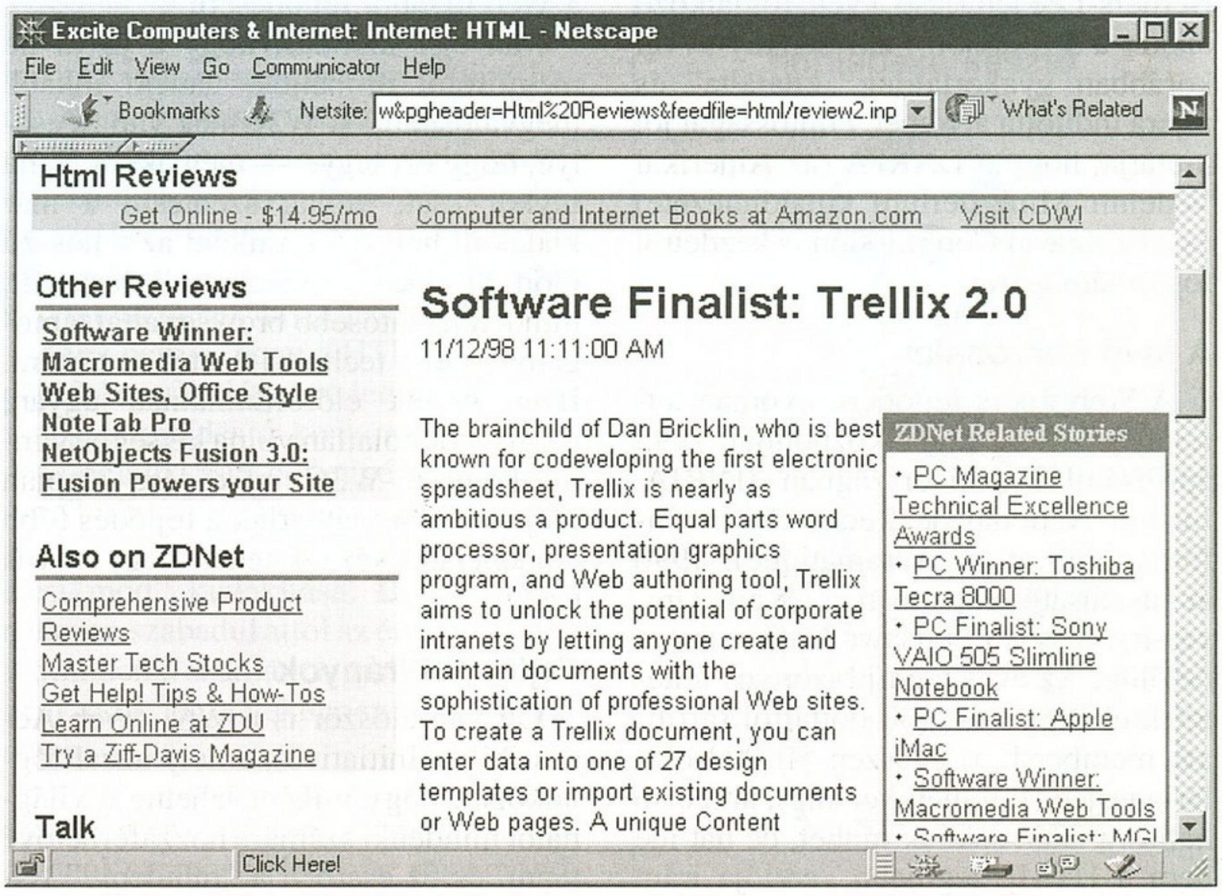
A WWW formálódását ma a nagy cégekből álló tömörülések határozzák meg (gondoljunk csak a Microsoftra vagy a Netscape+AOL felállásra), és a magánszemélyek szerepe háttérbe szorult. Nyugodtan mondhatjuk, hogy nem nekünk, nem a kisembereknek fejlesztik a Webet. Ez pedig pozitív és negatív következményekkel egyaránt járhat. A pozitívum főleg az, hogy fejlődik a hardver és a szoftver, ezáltal mindenféle új alkalmazás (benne film, televízió, hang stb.) a Weben keresztül is könnyen használhatóvá válik. A negatívum pedig főleg az, hogy a kommercializálódás éppen a Web lényegét veszélyezteti: ami eredetileg az ötletek szabad csereberéjét, a tudásvágy kielégítésének lehetőségét, az információk ingyenes közzétételét szolgálta, abból százmilliós nagyságrendű fogyasztói tömeget kiszolgáló online világáruház lesz. E két koncepció között pedig óriási a szakadék.

Ennek ellenére Berners-Lee nem fél a jövőtől: szerinte bőven van hely mindenféle kultúrának, és abba még a kereskedelem is belefér. *„...Amikor a Weben a piac igényeit már megfelelő mértékben kielégítik (ez valószínűleg pár év múlva következik be), a Web ismét készen áll majd az egyénnek mint személynek, nem pedig fogyasztónak a kiszolgálására, és egy új, valóban interaktív, a kreativitást ösztönző médiummá alakulhat, ami már nemcsak az információáramlást segíti majd, hanem az önkifejezés táptalaja is lesz, és külön életteret hoz létre..."* — mondja. Ami pedig a szórakoztatóipart illeti, ott a számítógép és a televízió még közelebb kerül egymáshoz — természetesen a televízió rovására. Aki pedig nyer a dolgon, az a felhasználó lesz, mert új hálózati szórakozások között válogathat.

De ez még mindig csak töredéke a Weben ránk váró jövőnek. *„...Ha létrehoztunk ilyen információs teret, amelyhez hozzákapcsolható egy nagymértékben feljavított ember–számítógép interfész, és ezáltal lehetővé válik például az intelligens beszédfelismerés, akkor világméretű számítógép-alapú rendszer jön létre, valamiféle globális agy, amelyben az egyes emberek a neuronok"* —jelentette ki Tim Berners-Lee.

Egyébként pedig bizakodjunk, hogy az eljövendő Web gyors lesz, kevesebb szemetet találunk majd rajta... Na jó, és esetleg még vásárolunk is!

Tóti Balázs



W3C — az istenek tanácsa

Konzorcium és konzílium

A World Wide Web Consortium (W3C) jelenleg minden bizonnyal a legfontosabb és legnagyobb befolyású webes szervezet. Tagjai között megtalálható a Microsoft, a Netscape és az America Online (most már együtt), egy sor nagybank, az IBM, az Apple, a Compaq és a többi jelentős szerepet játszó cég — tényleg mindazok, amelyek számítanak, illetve amelyeknek számít, hogy mi lesz a Webbel a jövőben. Azt pedig egyre kevesebben engedhetik meg maguknak, hogy ne foglalkozzanak ezzel a témával.

Az Internet, amely néhány ezer hozzáértő, technikai beállítottságú, leginkább angolul beszélő és hímnemű szakember házipostájaként indult, csak 1993 őszén vált széles körben hozzáférhetővé, amikor egyik pillanatról a másikra számos amerikai egyetem is rákapcsolódott. Nyugaton (elsősorban persze Amerikában) a könyvesboltokban már egy évvel később külön „Internet" polcok voltak.

1994 októberében sor került a World Wide Web Consortium létrehozására is. Ennek a „nemzetközi ipari egyesületnek" eredetileg a Massachusetts Institute of Technology Laboratory for Computer Science (USA) adott otthont. A W3C célja a kezdetektől fogva az volt, hogy „hozzásegítse a World Wide Webet lehetőségeinek teljes kibontakozásához a szükséges protokollok fejlesztése révén". Vezetője az a Tim Berners-Lee lett (lásd a vele foglalkozó cikket a 5. oldalon), aki néhány évvel korábban gyakorlatilag „kitalálta" és útjára indította a Webet. Fontosságát jól mutatja, hogy a DARPA (az Amerikai Védelmi Minisztérium kutatóintézete) és a European Commission is kezdettől fogva támogatta.

A jövő körvonalai

A Web gyors fejlődése nyomán létrejöttek a W3C újabb központjai: 1995 áprilisától Franciaországban (INRIA, Institut National de Recherche en Informatique et en Automatique), 1996 augusztusától Japánban is (Keio University, Shonan Fujisawa Campus) van belőlük. Az éves tagdíj bizonyos feltételektől függően 5000 dollártól (affiliate membership) egészen 50 000 dollárig terjed (full membership), ami első hallásra talán soknak tűnhet, de hát ide nem a HTML legújabb verziója iránt érdeklődő amatőrök iratkoznak be, és nem is lenne erre lehetőségük, mert a konzorcium a magánszemélyeket udvariasan arrébb tessékeli. Ám nekik sem árt, ha ismerik a W3C honlapjának címét: www.w3.org), mert onnan a Webbel kapcsolatban sokféle információ beszerezhető, a legújabb specifikációktól kezdve a HTML várható fejlődési tendenciáira vonakozó előrejelzésekig. Az pedig valóságos aranybányává teszi, hogy ott vannak a referenciául szolgáló kódok implementációi, és onnan tölthetők le a W3C által kifejlesztett új technológiákat bemutató és megvalósító szoftverek (amilyen a mostani CD-mellékleten a Linux és Windows alatt futtatható Amaya browser és HTML-szerkesztő). Aki ide ellátogat, az nem fog unatkozni, mert majdnem biztos, hogy a konzorcium honlapján felbukkanó kérdések kihatással lesznek a Web további fejlődésére.

Van egy mondás, hogy a jövőt ne megjósolni próbáljuk, hanem inkább megvalósítani. A W3C-nek van rá esélye, hogy ezt tegye — még akkor is, ha olykor csak utólag szentesíti a már kialakult helyzetet, miként az a hosszú időn át „nem hivatalosan" használt, minden jelentősebb browser által támogatott frame technika esetében történt. Ilyen esetek előfordulhatnak ugyan, mégis indokolatlan volna kétségbe vonni, hogy a W3C működése alapján idejében előre láthatjuk a fejlődés főbb tendenciáit. Nézzük tehát, hogy mik is ezek.

Fejlődési irányok

Ott van először is a WAI (Web Accessibility Initiative), amely azzal foglalkozik, hogy miként lehetne a világhálót mindenki számára hozzáférhetővé tenni, és ez esetben a „mindenki" felöleli a különböző fogyatékosságokkal küszködőket is. A W3C által kidolgozott specifikáció — mondta 1998 közepén a program vezetője, Judy Brewer — „útmutatót ad a webböngésző programok és a multimédia lejátszók fejlesztőinek ahhoz, hogy a fogyatékosok által is használható termékeket tudjanak előállítani". Ezáltal — teszi hozzá Scott Marshall, az Amerikai Vakok Szövetségének elnökhelyettese — hamarosan „a vakok és a csökkent látóképességűek ugyanúgy képesek lesznek navigálni a Weben, mint egészséges szemű társaik". Ez a tendencia már a HTML 4.0-s változatában is felfedezhető (lásd a HTML múltjáról és jelenéről szóló cikket a 17. oldalon), és lényegében arra vezethető vissza, hogy a Web használatának tömegessé válásával az elvileg csak néhány százalékot érintő problémák is a felhasználók százezreinek, sőt millióinak életét keseríthetik meg. Csak egyetlen analógia: a 20. század elején még el lehetett várni a levegőbe emelkedésre vágyóktól, hogy atletikus képességekkel rendelkező mérnök-ezermesterek legyenek. Ma már a légitársaságoknak a mozgássérültek utaztatását is rutinból meg kell oldaniuk.

A tényleg tömegessé váló Weben (márcsak üzleti megfontolásokból is) ugyanúgy oda kell figyelni a fizikailag fogyatékosokra, mint mondjuk azokra, akik matematikai formulákat szeretnének felrakni weboldalukra, és ehhez a postscript leírónyelvet nem találják kielégítő megoldásnak. Jött tehát a W3C, és közreműködésükkel XML-ben megírták (lásd az erről szóló cikket a 20. oldalon) a jelenleg W3C ajánlásként elfogadott Mathematical Markup Language, vagyis MathML 1.0-s változatát. Ez kétféle címkét (tag) tartalmaz: az egyik a formula vizuális megjelenítését szolgálja, a másik a tulajdonképpeni jelentést „címkézi".

A konzorcium szakemberei szerint az egész nem arra lett kitalálva, hogy „közvetlenül" írjunk vele a képernyőre, hanem speciális szoftvert kell hozzá használnunk. Korábban könnyedén megtehettük, hogy amennyiben eléggé ismertük a HTML-t, akkor jó idegekkel és edzett szemekkel akár sima karakteres szövegszerkesztővel megírjuk hon-

lapunkat. Ennek azonban (ha nem ma, akkor holnap) egyszer vége lesz. Dehát — valamit valamiért — ennek az az ára, hogy amennyiben bonyolultságot, flexibilitást és hasonlókat akarunk, akkor egy bizonyos szinten túl le kell mondanunk az igazán egyszerű eszközökről. A jövőbeni igényesebb (vagy például matematikával terheltebb) webes felületeket egyre kevésbé lehet majd a Norton Editorral összetákolni. Ugyanez a folyamat zajlott le a szövegszerkesztőknél is: az ún. átlagfelhasználó egészen biztosan nem tudja, hogy egy WinWord dokumentumban hogyan lesz a kiemelt szöveg kövér vagy dőlt betűs. És nem is érdekli. Neki — érthető módon — az a fontos, hogy minden különösebb nehézség (sőt, minden különösebb odafigyelés) nélkül a saját anyanyelvén is tudjon dolgozni.

Nemzetköziség

Ezzel máris újabb problémához érkeztünk. A W3C ezt internacionalizálódásnak, vagyis nemzetközivé válásnak nevezi, 1994 óta foglalkozik vele, és azt érti rajta, hogy mivel a Webet a világon mindenütt egyre inkább használják, növekvő igény van arra is, hogy az ember saját, többnyire speciális karaktereket használó nyelvén, például arabul, héberül, hindiül, koreaiul, kínaiul, japánul, thai nyelven (vagy hogy a dolgot ad abszurdum vigyük: magyarul) publikálhasson a Weben. Jelenleg ehhez az ISO 10646-os Unicode szolgál alapul a HTML 4.0-s verziójában is, lehetővé téve, hogy egy-egy bekezdést más nyelven írjunk meg, vagy a helyi szokásoknak megfelelően egy telefonszám jobbról balra legyen olvasható (a hagyományos örmény számokról és a japán kanáról nem is szólva).

A speciális nyelvi követelmények technikai megvalósítása persze közel sem olyan egyszerű, mint szeretnénk, hiszen egy európai browsernek miért is kellene megküzdenie a talán soha nem használt keleti nyelvek karaktereivel. Hasonló problémák merülhetnek fel a különböző VRML (vagyis a Virtual Reality Modeling Language) megoldásokkal kapcsolatban a webes alkalmazásoknál. Jelenleg úgy tűnik, hogy a legígéretesebb megoldás egy vagy több — minél inkább platformfüggetlen — fontspecifikáció volna. A meghatározott helyeken (értsd: szervereken) tárolt karakterkészleteket az Interneten keresztül mindig le lehetne tölteni az adott dokumentumot olvasó böngészőbe.

Ez azonban újabb kérdésekbe torkollik. A technológia ugyan jelenleg is lehetővé teszi, hogy akár töredékcenteket fizessünk egyes szolgáltatásokért, ma még általános gyakorlat, hogy azért is fizetünk, amit nem használunk, például az egész zenei CD-t megvesszük, pedig arról csak egy bizonyos számra van szükségünk. A jövőben ez biztosan változni fog, megvehetjük vagy bérelhetjük a részleteket is. Sőt, fizetni is csak annyit kell majd érte, ahányszor meghallgattuk (ahogy az áramszolgáltatónak is csak a fogyasztás arányában fizetünk). Alkalmasint az Interneten használt speciális fontokkal hasonlóképpen lesz. A szakértők egyelőre még vitatkoznak aazon, hogy kinek kell állnia a költségeket a látogatók számának arányában: annak, aki a weblapot készítette és működteti, vagy pedig a weblapot felkeresők fognak a szolgáltatásért apró összegeket átutalni a „fontszerver" fenntartójának. Ráadásul egyes betűtípusok szabadon felhasználhatók lesznek, mások pedig nem.

Emberi tényező... meg hatalmi

A technika önmagában persze kevés, és foglalkozni kell azzal is, hogy miként hat a társadalomra. A konzorcium 1997-ben létrehozott egy munkacsoportot ennek tanulmányozására. A webes technikai, illetve társadalmi problémák persze meglehetősen sajátosak: gondoljunk csak arra, hogy a külöböző országok az éppen érvényes törvényeknek megfelelően mennyire különbözőféleképpen ítélhetnek meg olyan technológiákat, mint a titkosító PGP. Tehát ilyesmire is tekintettel kell lenni, amikor általános, az egész Web működését, sőt jövőjét befolyásoló döntéseket hozunk. A W3C szerint úgy kell lehetővé tenni a helyi ellenőrzés gyakorlását, hogy az ne vezessen a helyi (vagy bármilyen más) kultúra dominanciájához. Ez ugyan nagyon szép elképzelés, de azért biztosan sokan eltűnődnek majd rajta, hogy a világ vezető számítástechnikai és egyéb hatalmasságaiból összetoborzódott szervezet mennyire lehet képes akár az ajánlások szintjén is dönteni ennek kivitelezéséről.

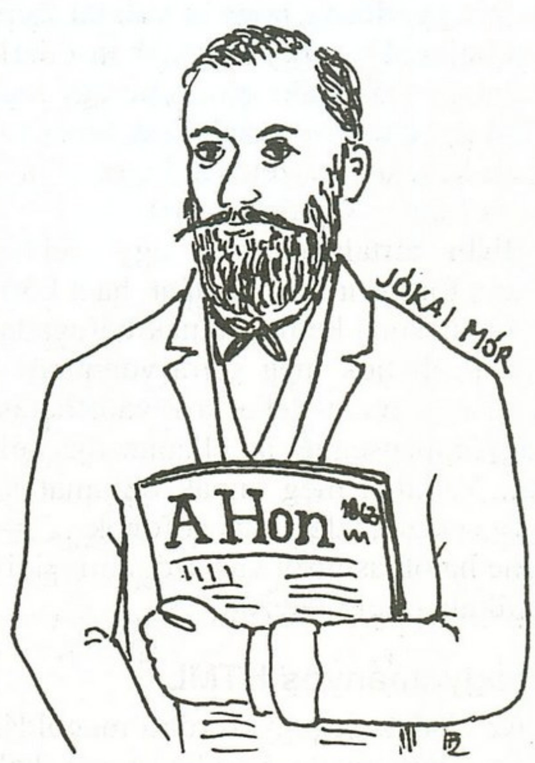
Tény persze, hogy a HTML esetében például egyáltalán nem lett volna lehetséges a fejlődés a konzorcium működése nélkül, és a cél egyébként is egy minél általánosabb és minél inkább mindenki számára megfelelő „együttélési mód" kialakítása. De az ember nehezen szabadul attól az érzéstől, hogy a „mindenki számára legjobb" fogalmát nem csak egyféleképpen lehet értelmezni.

A konzorcium elképzelései mindenesetre határozott irányba mutatnak, és olyanok szerepelnek benne, mint mondjuk a „metadata", amely úgy fogható fel, mint „adatról szóló adat" vagy még inkább „információról szóló információ". Ott van továbbá a digitális aláírás továbbfejlesztése, méghozzá úgy, hogy az eddigi gyakorlattal ellentétben az ne annyira az embernek, mint inkább a gépnek szóljon. A tervek szerint a jövőben a digitális aláírások ellenőrzése is a számítógépekre hárul majd, a kérdés csak az, hogy milyen körben lesz rá igény. A vásárlásnál persze igen, és ennek támogatása a jelek szerint az egyik elsőrendű feladat: „a Web növekedésében alapvető tényező az elektronikus kereskedelem: az eladás, a vásárlás és a hirdetés" — állapítja meg a W3C.

Valószínűleg túlzás lenne azt állítani, hogy a csökkent látóképességűek webhasználatának elősegítése közvetlen kereskedelmi megfontolásokra vezethető vissza, de tény, hogy ezáltal is újabb százezrek férnek hozzá a Webhez, ami sem az internetes szolgáltatóknak, sem a szoftvergyártóknak, sem a telefontársaságnak nem hátrányos. De ezen nem is kell csodálkozni. Az Internet „grafikus felhasználói felületűvé" és tömegessé válásával könnyen fel lehetett fedezni a Webben rejlő üzleti lehetőséget. És még azt sem mondhatjuk, hogy az elektronikus kereskedelem rosszabb (vagy éppen jobb) volna, mint a „valódi". Ugyanolyan. A különbség legfeljebb annyi, hogy ennek eredményes működtetéséhez időnként HTML-ajánlásokat kell kiadni, és általános alkalmazhatóságáért egyet és mást még meg kell tenni.

Galántai Zoltán

Az első magyar honlapszerkesztő



Nem mindenki születik művésznek

Csicsa és technika

A Clinton elnök szexbotrányát politikai kabaré gyanánt elénk táró multimédiás, online szappanopera egyes epizódjait már úgy rendezték meg, hogy a rágcsálnivaló információk közreadásában az Internet minden más közlési eszközt megelőzzön. Ha így megy tovább — és a sávszélesség istene is úgy akarja —, a hollywoodi látványgyárosok hamarosan felfedezik a Web műfaját.

Az alkalmazott művészetek legújabb ágának bölcsőjénél egyelőre főleg amatőrök tolonganak, akik sorra kipróbálják, hogy mit tudnak produkálni a csecsemő W3C feliratú csörgőjével. Mi jut eszünkbe arról a szóról, hogy csicsás? Olyasmi, hogy agyondíszített, felcicomázott. Lehet személy, aki teleaggatja magát mindenféle mütyürrel, bizsuval, össze nem illő ruhadarabokkal, de lehet a „mű" tárgy is, például egy autó, amelyre „egész estét betöltő" női alak van rápingálva. És sajnos több százezer, sőt, talán több millió weblap is ilyen csicsás. Hogy mitől az? Ezt próbáljuk meg elemezni az alábbi írásban.

A cégek számára presztizst jelent a honlap. A technikai lehetőségek ismeretében pedig nem csupán valami statikus, végső soron egy magazinhoz hasonló információközlést és megjelenést várnak el tőle, hanem az érdeklődést felkeltő multimédiás egyveleget is, amely látványos és szórakoztató, megragadja a figyelmet, felkelti az érdeklődést stb. Vegyük észre, hogy ebben már nem a tartalom játssza a főszerepet. Emberi gyarlóság, hogy ha valami szép, arra hajlandók vagyunk jobban odafigyelni és többet áldozni, mint egy legalább ugyanolyan tartalmú és minőségű, de kevésbé tetszetős dologra — lásd Linux kontra Windows :-)))

Hiába tartalmaz tehát egy weblap igazán fontos információkat, ha a körítés nem csinál kedvet annak befogadására. Próbáljuk meg sorra venni röviden, hogy mivel lehet a kívánt hatást (vagy éppenséggel az ellenkezőjét) elérni. Vállalva még annak ódiumát is, hogy persze „ízlések és pofonok..." — pláne ha olyasmiről van szó, ami eddig egyáltalán nem létezett.

A hagyományos HTML

Az első és legegyszerűbb megoldás a képek használata, és ehhez nem is kell különösebb tudás. Képzeljük csak el, hogy a háttér egy halovány tájkép, amelyen élvezet a sötét tónusú szöveget olvasni, az egész megnyugtató, szempihentető. Persze ennek is megvan a hátulütője, mert a figyelem felkeltésére viszont kevésbé alkalmas. Ahhoz valami meghökkentő, provokatív dologra van szükség, amilyen például egy sztroboszkópos hátterű animáció, vagy ügyesen megkomponált frame-ek szokatlan kompozíciója.

A frame-ek használata egy kicsit több tudást, vagy legalábbis egy jó HTML-editort igényel, de ha bármelyik megvan, akkor viszonylag könnyű olyan vizuális elemeket létrehozni, amilyeneket frame-ek nélkül elég nehézkesen lehetne. A frame ez esetben ugyanis azt jelenti, hogy a felhasználó által a monitor képernyőjén látott oldal tulajdonképpen több oldalból tevődik össze. Az egyiken mondjuk egy embléma (logo), a másikon egy menüsor, a harmadikon az érdemi információ található. (Lásd http://www.intruder.hu. Itt és a későbbiekben szemléltetésül leginkább az általunk készített weblapokra fogunk hivatkozni, mert azoknál pontosan meg tudjuk mondani, hogy milyen megoldás milyen megfontolásokból született.)

A következő lépés a multimédia. Ma már az animáció és a zene lehetőségeit is kihasználhatjuk a weblapokon. Persze csak módjával, és egyáltalán nem biztos, hogy érdemes. Az animáció elkészítése nem is tartozik a legegyszerűbb dolgok közé. Az adott szoftver kezelésének ismerete mellett némi ötletesség is kell hozzá, mert egy vigyorgó és egy szomorú pofa váltakozása egy idő után senkit nem nyűgöz le. (Kezdeti ujjgyakorlatnak persze nem rossz.) Ha kézügyességünk erősen konvergál a zéró felé, akkor animáció helyett viszont célszerűbb valami más megoldással próbálkozni, leginkább talán a már meglévő mozgóképek Webre történő digitalizálása ajánlható.

A zenét legegyszerűbben MIDI fájlok formájában írhatjuk bele honlapunkba. De még mielőtt ezt tennénk, gondoljuk végig, hogy ez esetleg éppenséggel idegesíteni fogja az arra tévedőket, különösen ha már sokadszor hallják. Maga a hangállomány viszonylag egyszerűen előállítható az arra alkalmas programokkal (Voyetra Sequencer, Midisoft Studio stb., de rengeteg található belőlük az Interneten is).

Ennyivel lényegében ki is merítettük az egyszerű, teljesen HTML szabványú honlapok (szerencsére vagy sajnos) nem túl gazdag csicsázási lehetőségeit.

Legalább középfokon

A Java applet és a JavaScript a két kulcsszó weblapjaink színesebbé, érdekesebbé tételéhez. Csinálhatunk rákattintáskor „benyomódó", háromdimenziós gombot, de olyasmit is, hogy amikor a kurzor egy meghatározott pont fölé ér, akkor az egyik képet a scriptnek köszönhetően felváltja a másik. Ez természetesen kiválóan használható idétlen bosszantásra is (van olyan weblap, ahol odakattintva a gomb egyszerűen eltűnik, és semmi más nem történik), vagy éppen reklámozhatjuk magunkat a böngésző státuszsorában megjelenített, szembántóan vibráló, „scrollozó" szöveggel.

A Java segítségével megjeleníthetünk olyan kockát (http://www.ienet.hu/hun/design/java/java.htm), amely az egér pozíciójától függően forog. Megoldhatunk rengeteg hasonlóan látványos, de több programozást igénylő feladatot is (http://www.jovonezo.hu/comp.htm). A Java nagy előnye a CGI scripttel szemben az, hogy azt böngészőnk saját gépünkön futtatja, és nem terheli a szervereket. De ez a dolognak csak az egyik fele. A másik az, hogy a JavaScriptet nem ugyanarra szokás használni, mint a CGI-t (http://www.digitel2002.hu/kuf).

Ha úgy döntünk, hogy a CGI-t hívjuk segítségül, és egy kicsit jártasak vagyunk a programozásban is (meg persze megfelelő jogokkal rendelkezünk a szerveren), akkor tényleg mindent megcsinálhatunk. Ott van például a számláló, a Weben alkalmazott leggyakoribb CGI-s megoldás: teljesen automatikusan naplózhatjuk, hogy mikor és honnan nézték meg a lapunkat. Jelszavas védelmet is adhatunk vele a honlapunknak. A dolog egyetlen hátulütője, hogy a szerveren fut, tehát működése a terheléstől függően idegesítően lassú is lehet. Mindent egybevetve azonban a

lehető legsokoldalúbb lehetőségeket nyújtja, és manapság a C-től kezdve, a Visual Basic-en át a Delphi-ig számos programozási nyelv támogatja ezt a formát.

Tehát csicsa vagy nem csicsa?

Bár a fenti eszközök könnyen arra csábíthatnak, hogy csicsás megjelenítést produkáljunk, a lényeg mégsem a technika. Egészen egyszerű módszerekkel is alkothatunk csicsás honlapokat, és bonyolult programozási eszközök latbavetése is eredményezhet elegáns, egyszerű és esztétikus megoldásokat. „Nyitott szemmel" szörfözve az Interneten, bármikor találhatunk figyelemre méltó ötleteket, sőt, vannak külön a weblapok fejlesztésére specializálódótt helyek is, ahonnan háttereket, animációkat stb. gyűjthetünk be. A lehetőségeket illetően a határ a csillagos ég. Illetve a böngészőnk, mert ha az valamit nem tud, akkor nincs tovább. Legfeljebb abban reménykedhetünk, hogy majd a következő verzió..., illetve hogy találunk a céljainknak éppen megfelelő pug-int (kiegészítőt). A kínálat ma már „csicsázási szempontból" is kimondottan gazdag: akad olyan plug-in, amelynek köszönhetően multimédiás szemüvegünkön át három dimenzióban tárul elénk a Világháló.

A HTML fejlődése további eszközöket ad a kezünkbe. A 4-es verziónak már része az ún. layer technológia: itt különböző, egymás felett mozgó síkokat lehet kreálni, és a síkok azután (feliratokkal vagy képekkel együtt is) a már említett JavaScriptnek köszönhetően mozgathatók egymás felett (http://www.net.hu). A rétegekben (layer) pedig sokféle HTML címke (tag) helyezhető el.

Vagy ott van a HTML lényegében továbbra is kétdimenziós világát valódi háromdimenziósra cserélő VRML (Virtual Reality Modeling Language), amelyben már mozgásra, sőt játékra is van lehetőség. Ha kellőképpen mértékletesek vagyunk (mint minden másban, a weblapkészítésben is ez a legjobban bevált alapelv), akkor ennek is megtalálhatjuk a helyét, és bátran használhatjuk. Arra azonban érdemes odafigyelni, hogy a jelenlegi adatátviteli sebességek mellett egy ideig még érdemes lesz takarékosan gazdálkodni a bitekkel meg a bájtokkal.

Az ésszerűség határain belül természetesen a divatot is követhetjük, ha esetleg mégis zavarna bennünket, hogy honlapunkat maradinak vagy megkövesedettnek tartják. Aki csak most tervez honlapot, az valószínűleg már figyelembe veszi az új stílust. Nem is olyan régen a nagyon sötét vagy nagyon kontrasztos grafika volt a megszokott. Újabban az egyszerű, pasztellszínű, háttérbe beolvadó világos képek váltak népszerűbbé (lásd http://www.nokia.hu vagy http://www.audi.com).

Jósolgatások

1999-ben új varázsszavak terjednek el: DHTML (dinamikus HTML), CSS (lépcsőzetes stíluslapok), XML (Extensible Markup Language). Nézzük meg legalább érintőlegesen, hogy mik is ezek.

— A DHTML a dinamikus tartalom szabványa. A dinamikus tartalom feldolgozását jobbára a böngészőre bízza, tehát az egész sokkal gyorsabb lesz, és sokkal kevésbé igényel kiegészítő modulokat. Ráadásul gazdagabb grafikai és adatkezelési lehetőségeket ígér, rövidebb letöltési időkkel, és különösen a felhasználó közreműködésétől függő megjelenítést egyszerűsíti le. A fejlesztőknek viszont igen sok fejfájást fog okozni, hogy a két meghatározó szerepű DHTML implementáció (Netscape és Internet Explorer) nem egyforma.

— A stíluslapokkal készült oldalak más dokumentumoktól örökölhetik megjelenítésük egyes részleteit. Jelenleg a CSS és a HTML kéz a kézben halad, de a jelek szerint az XML-nek is lesznek saját stíluslapjai (az Extensible Style Language, röviden XSL).

— Az XML-es dokumentumok (a részletekről lásd a 20. oldali cikket) tartalmazzák a keresett információra vonatkozó kódokat. Az adatellenőrzés nagyrészt a böngészőre hárul, ahogy gyakran a dinamikus tartalom kezelése is. Ez a nyelv a HTML-lel ellentétben a megjelenítés részletei helyett a dokumentum struktúráját írja le, így az eddiginél sokkal több rugalmasságot biztosít a fejlesztőknek. A HTML és az XML azonban ennek ellenére nem helyettesíti, hanem inkább kiegészíti egymást. 1996 szeptembere, a fejlesztés megkezdése óta az XML rohamosan tör előre, és immár nem kétséges, hogy a Világhálón ez fogja az SGML alapú publikálást hordozni. Méghozzá leginkább az alábbi négyféle alkalmazásnak köszönhetően:

— Olyan alkalmazások, amelyeknél két vagy több heterogén adatbázis közötti kapcsolat létrehozásához szükség van a kliensre.

— Alkalmazások, amelyek a szerverről a kliensre akarják áttenni a feldolgozási műveletekkel járó terhelés jelentős részét.

— Olyan alkalmazások, amelyekben az adathalmazt a kliensnek kell megjelenítenie, méghozzá különböző felhasználók számára különböző formában.

— Olyan alkalmazások, amelyeknél majd intelligens ügynökök döntenek a felhasználók adatvisszakeresési igényeinek teljesítéséről.

Mindez, valljuk be, roppant csábítóan hangzik, ám az XML túlságosan új, és túlságosan eltér a megszokott HTML-től, így sok fejlesztő fog még keservesen szenvedni a nagyobb rendszerek konvertálásával és a munkatársak felkészítésével az új nyelvre. Talán érdemesebb is lenne a HTML-t újabb eszközökkel kibővíteni, a régi szintaxisok nagy részének megőrzésével. A HTML ugyanis nem halt meg, csak saját sikerétől szenved. De szenved a felhasználó is, amikor a „404 URL not found" hibaüzenetet olvassa. Hozzá kell tehát látnunk az egyszerű HTML továbbfejlesztéséhez — vagy lecseréléséhez. Aminek következtében persze a csicsás, értelmetlen figyelemfelhívás új, eddig soha nem látott válfajai is meg fognak jelenni. Ez sajnos a dolgok velejárója. A mai nyomtatott sajtó jelentős részéről is nehéz lenne feltételezni, hogy rokoni szálak fűzik a tipográfia művészetéhez. Pedig az a családfa már több mint 500 éves.

Balla László – Borsányi Attila



Megtehető, de...

Apró kis gonoszságok

Alig egy-két éve még csak meglehetősen egyszerű eszközök álltak a webtervezők rendelkezésére, így azokkal viszonylag szimpla túlkapásokat tudtak elkövetni a designerek. Ilyen volt például a Netscape által bevezetett „blink", vagyis a szöveget idegesítően villogtató címke. Mára alaposan megváltozott a helyzet, de aki igazán látványos vagy éppenséggel „sokkoló" megoldással akar kirukkolni, annak bizony bele kell egy kicsit mélyednie a JavaScript programozásába.

Az alábbiakban a BUILDER.COM által legelterjedtebbnek talált 10 játékos, öncélú vagy bosszantó „fogást" mutatjuk be, a trükkök konkrét technikai ismertetése nélkül. Vannak gonoszabbak is, de nem propagáljuk őket.

Elöljáróban még annyit, hogy lassanként már eljutunk oda, hogy a témának ne csupán pszichológiája és szociológiája, hanem patológiája is legyen.

1. Jelszóvédelem

Ha a weblapunkra feltett információt nem akarjuk mindenki számára hozzáférhetővé tenni, akkor számíthatunk rá, hogy a legtöbb szerver támogatja az ún. „Basic Authentication" funkciót, és még akkor is jelszót rendelhetünk egy meghatározott folderhez (a kívülállók heccelésére), ha nincsenek rendszergazdai jogaink. Elég, ha írási joggal rendelkezünk az adott helyen.

2. Óra

Egy meglehetősen egyszerű JavaScript segítségével kiírhatjuk a webhelyre a helyi időt. Mert miért is ne? A BUILDER.COM szakértői ezzel kapcsolatban meglehetősen gúnyosan így fogalmaznak: „Hiszen mi is volna hasznosabb, mint a pontos idő és a dátum! Pont ez a fajta információ az, ami online hozzáférhetővé téve megváltoztatja az életünket."

3. Ideje, hogy...

Ha már felszereltük honlapunkat a pontos idő digitalizált kakukkos órájával, akkor JavaScript segítségével azt is beépíthetjük, hogy reggel 8 és 12 között egy felirattal jó reggelt kívánjon az oda látogatónak, este 10 után pedig arra kezdjen célozgatni, hogy ideje volna nyugovóra térni. Hátha a szerencsétlen nem jön rá magától.

4. Számláló

Ez már nem okvetlenül haszontalan, főleg, ha megfelelő helyen használjuk. A „counter", azaz számláló arról szolgáltat információt (alkalmasint részletes statisztikával is), hogy az adott helyet hányan keresték fel. Igaz ugyan, hogy internetes szolgáltatónk is tudna ilyesmit prezentálni, és vannak ingyenes vagy majdnem ingyenes külső számlálók is, ám ezek néha csigalassúsággal töltődnek be a távoli gépről. Az sem elképzelhetetlen, hogy egy hibásan megírt Java programocskát (appletet) futtatva lelövik látogatónk böngészőjét. Tehát mindenképpen érdemes alaposan megfontolni az igénybevételét.

5. Testre szabott háttér

A JavaScriptnek köszönhetően azt is felajánlhatjuk a honlapunkra látogatónak, hogy válassza ki saját maga a háttér színét, hátha sárga vagy világoskék alapon szeretné olvasni a szöveget. Ez megint csak nem valami jó ötlet, mert a hátteret online átállítgatók másokat is feltartanak a Weben.

6. Fényújság

JavaScript segítségével jobbról balra folyamatosan mozgó szöveget írathatunk a browser ablaka alatti státuszfejléc keretébe (scrolling text). Ezzel „kvázidinamikus" kinézetet kölcsönözhetünk weboldalunknak, mivel azonban a legügyesebb JavaScript sem gondoskodik róla, hogy valami értelme is legyen a dolognak, jobb, ha inkább tartózkodunk tőle.

7. Vendégkönyv

Ez már kissé bonyolultabb. Olyan CGI-scriptről van szó, amely egy HTML-formból kapja az inputot, és amely által a látogatónak lehetőséget adunk, hogy közvetlenül írjon a weblapunkra. Számos Internet-szolgáltató nem is engedi meg ügyfeleinek, hogy egy ilyen vendégkönyvet (guestbook) működtessenek, mert az esetleg védelmi hibákat eredményezhet. Adott esetben kerülőút lehet viszont, hogy linket építünk ki valamelyik vendégkönyvet is szolgáltató ingyenes szerverhez. Persze csak akkor tegyük, ha komolyan szükségünk van rá.

8. Stáblista

Nem olyan gyakori, mint az eddigiek, de elég kellemetlen tud lenni ahhoz, hogy mégis megemlítsük. A JavaScript jóvoltából az általunk fontosnak tartott dolgokat egy „stáblistán" felsoroljuk: a megfelelő linkre kattintva beugrik egy kicsiny ablak. Az eredmény dinamikus. És határozottan idegesítő.

9. Tolakodó ablak

Megtehetjük, hogy amikor a felhasználó rákattint egy linkre, akkor előbb egy ablak jelenik meg valamilyen üzenettel (akár reklámmal is), és csak ennek az ablaknak a becsukása után tud továbblépni (Red Alert). Természetesen ez is JavaScript, lassítja a böngészést, és roppant bosszantó.

10. Átváltozó gombok

Végül ott vannak a gombok. Ha egynél több oldalból áll a webhelyünk, a navigáció igen fontossá válik. Sokan JavaScriptet használnak arra, hogy a kurzornak a gomb fölé kerülésekor megváltoztassák a gomb színét, alakját stb. Ismét az önmérsékleten és jó ízlésen múlik, hogy ezzel mennyire törekszünk borsot törni mások orra alá.

Két tanulságot mindenképpen érdemes leszűrni a fentiekből. Az első az, hogy mások froclizására kitalálhatunk ugyan „szellemes" dolgokat, de tudnunk kell, hogy ilyenkor mindig bumerángot dobunk el, amely jól kiszámítható röppályán visszatérve saját weblapunk „ablakát töri be". A második tanulság az, hogy miként az élet egyéb területein, a Weben sem érdemes mindazt megtenni, amire technikailag a lehetőség megvan.

Galántai Zoltán

WYSIWYG vagy kódszerkesztés

HTML-editorokat böngészve

A cikkben szereplő HTML-szerkesztők az Interneten található statisztikák alapján, de tagadhatatlanul szubjektív alapon lettek kiválogatva. Törekedtünk a (viszonylag) könnyű kezelhetőségre és az elterjedtségre. A futtató rendszerkörnyezet is lehetett volna OS/2 vagy Linux, de Macintosh, sőt DR-DOS és BEOS is, mégis a Windows 3.x és a Windows 95 melletti döntés a reális, mert egyelőre ezeket használják a legtöbben.

Ha valaki elkezd kacérkodni a Webbel, és annak egyelőre vitathatatlanul legfontosabb építőanyagával, a HTML felülettel, akkor egy bizonyos pontra eljutva általában kedve támad megírni legalább egyetlen „kísérleti" honlapot. Kérdés, hogy milyen HTML-editorral érdemes nekivágni ennek a kalandnak.

A mostani CD-mellékletre felrakott webszerkesztők lelőhelye általában a Yahoo, a Tucows (magyar tükörszervere is van) vagy a Davecentral. A cikk megírása idején, azaz 1998 végén érvényes helyzetet mutatjuk be, de a CD-re (annak későbbi „lapzártája" révén) frissebb verziók is felkerülhetnek. Még annyit a bevezetéshez, hogy az összefoglaló táblázatban szereplő szoftverek majdnem mindegyike időkorlátos (time limit), tehát csak 30, 60 vagy 90 napig működik, utána vagy regisztráltatjuk, vagy elbúcsúzunk tőle.

Azt kapod, amit látsz

Első megközelítésben könnyen felfedezhetjük a webszerkesztők alapvetően különböző koncepciójú két nagy csoportját:

— Az egyik kategóriába a weblapot (majd) a böngészőben is látható módon megjelenítő szerkesztők tartoznak (WYSIWYG, What You See Is What You Get, azt kapod, amit látsz). Ilyen például a Netscape beépített editora is. (E cikkben a programok neve mellé nem írjuk ki a verziószámot és az operációs rendszert, mert ezek az adatok megnézhetők a mellékelt táblázatban.)

— A másik nagy csoportba tartoznak az ún. kódszerkesztők, amelyekben egyszerű karakteres formában kell a HTML-kódot megírni, és az egész anyagot forráskódszövegként látjuk, a végtermék webes kinézetét külön valamelyik böngészőben kell megnéznünk.

A harmadik — és egyre népszerűbb — irányzat a fenti két koncepció ötvözése magában a webszerkesztőben. Ilyenkor a WYSIWYG oldalt és a kódszöveget egyaránt láthatjuk, sőt tetszés szerint bármelyiket szerkeszthetjük.

Kezdjük a WYSIWYG-gel

Ezek tulajdonképpen nagyon hasonlítanak egymásra. Inkább csak opcióik mennyiségében különböznek, és abban, hogy az adott nyelvet (a HTML 3.2-es vagy 4.0-s változatát) mennyire tudják kihasználni. Bármelyikkel is dolgozunk, közös bennük, hogy szerkesztéskor azonnal látjuk, milyen lesz az oldal egy böngésző ablakában. Vagyis az egész nagyon hasonlít egy grafikus szövegszerkesztő (például WinWord) működéséhez, amelyben szintén nem látjuk a formázóutasításokat, csak magát a megszerkesztett dokumentumot. Nem meglepő tehát, hogy a legelterjedtebb szövegszerkesztőkhöz elkészítették a HTML-esítő makrókat is. Az Office 97-ben lévő WinWord például menükből, űrlapszerűen képes HTML-lel dolgozni, bár hozzá kell tenni, hogy meglehetősen nyögvenyelősen.

A WYSIWYG kategóriában a „legbutább" minden bizonnyal a Netscape editora. Nagyon kevés opciót ismer, így komolyabb munkához nem ajánlható. Csak egészen egyszerű, „fapados" táblázatokat tud készíteni, nincs meg benne a formula (form), a keret (frame) és a script szerkesztésének lehetősége, tehát inkább csak azoknak ajánlható, akik nagyon egyszerű anyagokat akarnak készíteni, a HTML lehetőségeivel pedig nem akarnak mélyebben megismerkedni.

A Tarantula egészen különleges megoldást használ. Itt mindent mezőkbe (area) kell bepakolni, és ezeket lehet kedvünk szerint mozgatni a képernyőn. Az elképzelés szellemes, megoldásában mégis sok a fogyatékosság: bonyolult kezelni, külön mezőkben kell elhelyezni a képet, a szöveget, a formulát stb., továbbá az anyagot HTML formátumban csak nyakatekert módon lehet elmenteni. Még nagyobb baj azonban, hogy csak adott képernyőméretre készíthetünk vele lapokat.

Az America Online editora, az AOLpress már komolyabb jószág. Itt meg lehet tekinteni magát a HTML-kódot is, sőt, szerkeszteni is tudjuk. Közepesen jó szoftver, előnye az ingyenesség, hátránya a kevés opció.

Az egyik legjobb WYSIWYG editor a Dreamweaver („álomszövő"). Nagyon sok opciót ismer, és szinte mindent lehet vele paraméterezni, ezáltal igényes weblapok előállítására is alkal-



mas. Az egyes elemekre ráállva egy új ablakban minden paramétert könnyen módosíthatunk. Ráadásul megfelelő kódszerkesztő is tartozik hozzá.

(Az AOLpress és a Dreamweaver már inkább az ötvözet felé hajlik, de alapvetően mégis „látványszerkesztők".)

Maga a pőre HTML

A webszerkesztők másik nagy csoportját alkotó kódszerkesztők rendszerint különleges opciókkal felruházott ASCII editorok. Általános tapasztalat, hogy ezekkel a karakteres editorokkal a HTML leggyakrabban használt elemei (képek, linkek) könnyen és jól paraméterezhetők, míg az egyéb összetevők (formák, keretek, táblák) formázhatósága a szoftver készítője által felállított rangsortól függően romlik. Bővebb magyarázat helyett egy konkrét példa: a CoolCat-ben a JavaScripthez és a Javához varázsló (wizard) van, a képekhez, táblázatokhoz elég sok opciót ismerő űrlap, a keretek viszont csak minimális mértékben paraméterezhetők az űrlappal...

— A legegyszerűbb HTML-szerkesztők az olyan hagyományos texteditorok, amilyen például a DOS Edit vagy a Windows Notepad. Ezekkel természetesen bármilyen weblapot meg lehet írni, csak kellőképpen járatosnak kell lennünk a HTML nyelvben. Szerkesztés közben persze érdemes időnként behívni szövegünket egy böngészőbe, hogy ellenőrizzük az eredményt. Ennek a módszernek két nagy hátránya van: az egyik, hogy szemünket nagyon igénybe veszi a HTML címkék kavalkádjának követése, a másik, hogy minél bonyolultabb a megszerkesztendő weblap, annál alaposabban kell ismernünk a HTML-t, és a felmerülő problémák megoldásához rendszerint külső segítséget kell igénybe venni, például más HTML-oldalakat visszafejteni, vagy valamilyen nyelvleíró dokumentációt szerezni. Kétségtelen, hogy nagyfokú megszállottság kell az ilyesfajta szerkesztéshez, cserébe viszont tényleg bármit meg tudunk vele csinálni, ami egyáltalán lehetséges.

— Egy kicsit kényelmesebb megoldás néhány kódszerkesztőben, hogy az egyes címkékhez (tag-ekhez) gombokat és űrlapot rendelnek hozzá. Igaz ugyan, hogy csak az alapvető címkékhez (és kevés paraméterrel), meg esetleg egy-két elemhez (képekhez, linkekhez) van űrlap (Anansi). Az ilyen szerkesztők rendszerint nem ismerik a komplex elemeket (kereteket, táblázatokat, scripteket stb.), vagyis az egyszerű címkék begépelésétől megmentik ugyan az embert, de komolyabb weblapot a HTML ismerete nélkül azért sem lehet velük írni, mert általában a help is nagyon gyenge, az inkább magának a programnak a használatával foglalkozik.

— A következő kategóriába az űrlapos megoldású editorok tartoznak. Tipikus képviselőjük a DiDaPro. Űrlapon lehet néhány paramétert beállítani, és a program az így generált részt rakja be a szövegbe. Inkább az összefüggő, kevésbé rugalmas részek elkészítésére alkalmas (táblázatok, keretek). Hasonlóképpen az sem számít előnyös vonásának, hogy az ilyesfajta megoldásoknál gyakran lehet ugyan a HTML globális paramétereit módosítani, de nem valami intelligens módon (a szoftver nem a meglévő BODY címkét írja át, hanem beteszi abba a frissen generált BODY címkét). Előnye — mert az is akad —, hogy az űrlapoknak köszönhetően nem kell túl sokat tudni a HTML-ről. További hátránya viszont, hogy az űrlapon keresztül meglehetősen kevés paramétert lehet beállítani, ezért a végeredmény legfeljebb közepes minőségű lesz.

— Eggyel feljebb lépve vannak az „űrlapeditorok", amelyekben nem csupán majdnem minden paraméter beállítható, hanem a Java és a JavaScript támogatása mellett megtalálhatjuk a különböző browserek specifikációit is, és menüből vagy űrlapból kiválasztható beépített függvények is rendelkezésre állnak. Itt sincs szükségünk elmélyült tudásra, mert a szoftver ismeri helyettünk a HTML-t, de mert a súgók viszonylag részletesen kitérnek a HTML-re, lehet belőlük tanulni.

— A programozás felé hajlók számára készült editorok — például a HotDog különböző változatai — űrlapok helyett listát kínálnak fel a szükséges címkék kiválasztására. Egyes esetekben még a nem hivatalos, kizárólag az egyik vagy a másik böngésző által támogatott kiegészítések között is tallózhatunk. In-

Név	Operációs rendszer	Típus	HTML verzió	Fájl-import	Beépített néző	Támogatások *					
						Frame	Form	Táblázat	JavaScript	Java applet	Egyéb**
Netscape 4.0	W3.x/W95	WYSIWYG	3.2	Van	Van	Nincs	Nincs	(2)	Nincs	Nincs	(2)
HTML Write	W3.x	Teljes oldal	?	Nincs	Nincs	Nincs	Nincs	Nincs	Nincs	Nincs	(1)
Anansi	W3.x	Teljes oldal	?	Van	Nincs	Nincs	(1)	(1)	Nincs	Nincs	(1)
WebExpress	W3.x	WYSIWYG	3.0	Nincs	Nincs	(2)	(2)	(4)	(1)	Nincs	(1)
DiDaPro	W3.x	Teljes oldal	3.0/3.2	Nincs	Van	Nincs	Nincs	(6)	Nincs	Nincs	(2)
Word 97	W95	WYSIWYG	?	Van	Nincs	Nincs	(6)	(6)	Nincs	Nincs	(2)
WebEdit Pro	W95	Teljes oldal	3.2	Nincs	Van	(6)	(6)	(4)	Nincs	(1)	(3)
HTMLpad	W95	Teljes oldal	?	Nincs	Nincs	(2)	(1)	Nincs	(1)	(1)	(2)
HotDog Pro 5	W95	Teljes oldal	4.0	Van	Nincs	(5)	(5)	(5)	(5)	(5)	(5)
Hippie	W95	WYSIWYG/ Teljes oldal	?	Van	Nincs	(6)	(6)	(6)	Nincs	Nincs	(3)
1-4-All	W95	Teljes oldal	3.2	Nincs	Van	(6)	(1)	(3)	(6)	Nincs	(5)
Tarantula	W95	WYSIWYG	?	Van	Nincs	Van	Van	Van	Van	Van	Van
CoolCat	W95	Teljes oldal	4.0	Nincs	Van	(2)	(3)	(3)	(6)	(6)	(3)
AOLpress	W95	WYSIWYG/ Teljes oldal	?	Van	Van	Nincs	Nincs	(2)	Nincs	(2)	(2)
Aardvark Pro	W95	Teljes oldal	?	Van	Nincs	(2)	(6)	(6)	Nincs	(6)	(6)
Dreamweaver	W95	WYSIWYG/ Teljes oldal	4.0	Van	Nincs	(6)	(6)	(6)	(6)	(6)	(6)

* A zárójeles számok jelölik a támogatás milyenségét: (1)Alapszolgáltatás címkék elhelyezésével. (2) Kérdőív minimális lehetőségekkel. (3) Kérdőív néhány opció beállításával. (4) Kérdőív az összes ismert HTML specifikációval. (5) Címkék és opciók listás kiválasztása és kombinálása. (6) Varázsló (wizard). ** Az egyéb opciók: kép, link stb.

kább a HTML-kód írásán, és nem a szofverrel támogatott weblapkészítésen van a hangsúly, ezért a programozók kényelmét szolgáló opciók is be vannak építve. (Például a kód könnyebb kezelhetősége érdekében színkiemelést lehet választani, van HTML szintaxiskontroll, és egyéb olyasmi, amire az űrlapos megoldásnál nincs szükség.) Ezek a programok azonban végső soron inkább a begépelést gépesítik, és akkor is nagy szükség van a HTML ismeretére, ha elég jó súgó van a programba beleépítve (lásd HotDog). Leginkább a programozási nyelvek editoraira hasonlítanak.

— Az utolsó kategóriát a varázslóval (wizard) felszerelt szerkesztők képezik. Ez leginkább a HTML-ben járatlanoknak kedvez, mert a (többnyire) logikus „varázspálca" segíti a különböző elemek behelyezését. A legtöbb esetben WYSIWYG kiegészítés is van hozzá, és a varázslóban rögtön látszik, hogyan fog kinézni az adott elem. A varázslók bonyolultsága programról programra változik, de az egyszerűbb elemeknél, amilyen például a vízszintes vonal, a szoftver csak a minimumot fogja felajánlani, tehát az egységnyi vastagságú vonalat.

Öszvérek

A két ellentétes póluson tehát a kezdőket előzékenyen (bár általában igénytelenül) kiszolgáló WYSIWYG szerkesztők, illetve az írást megkönnyítő, de sem űrlapot, sem varázslót nem használó (csak az eredmény szemügyre vételéhez esetleg többféle browsert is behívni képes), inkább a programozói alkatúaknak való editorok helyezkednek el.

A kettő közötti „szürke zónában" a fenti két kategória képességei keverednek, de ezek a programok még nem alkotnak markáns önálló csoportot. Ilyennek tekinthető többek között a már említett AOLpress (kódszerkesztésre is képes WYSIWYG editor), vagy a Hippie (WYSIWYG ablakkal rendelkező kódeditor). Mindkettőjük megkövetel bizonyos nyelvismeretet, de a leggyakoribb elemekre van űrlap és varázsló. Ez a kategória azoknak ajánlható, akik szeretnének a sablonokon túllépni, de nem akarnak nagyon belegabalyodni a kódeditálásba.

Hogy a jövőben mi várható? A HTML-szerkesztők egyes kategóriái markánsabbá válnak (WYSIWYG, programozói editor, öszvér), de ezek közül meggyőződésem szerint a középutas megoldás lesz a legsikeresebb.

Tóth Csaba

Hatékony kompromisszum

Profi szerkesztők

A webszerkesztés első időszakában a hozzáértők valamilyen egyszerű texteditort használtak, mert csak 20-30 címkét kellett kezelniük, és azok beírását is meg lehetett könnyíteni a jól definiálható billentyűzetmakrókkal. Ma már azonban egyre bonyolultabb szerkezetű weblapokat kell készíteni, amelyek áttekintésére a kódolt szövegek dzsungele nem a legalkalmasabb. Találni kell új megoldásokat.

Az igazi profik a HTML írásánál még egy-két évvel ezelőtt is ragaszkodtak az Emacs-hoz, a vi-hez meg a kiválóan programozható, leginkább unixos környezetből származó többi editorhoz.

Nyilvánvaló, hogy a laikus weblapkészítőknek WYSIWYG editorra van szükségük. Bonyolultabb weblapok készítésekor azonban a profik sem mondhatnak le a produktum menet közbeni vizuális ellenőrzéséről, és ha ehhez túl sűrűn kell ingázniuk a kódszerkesztő és a böngésző között, az nem éppen hatékony megoldás. Lehetőleg egyetlen programon belül, párhuzamosan kell tehát megoldani a kódhoz való közvetlen hozáférést és a WYSIWYG kontrollt.

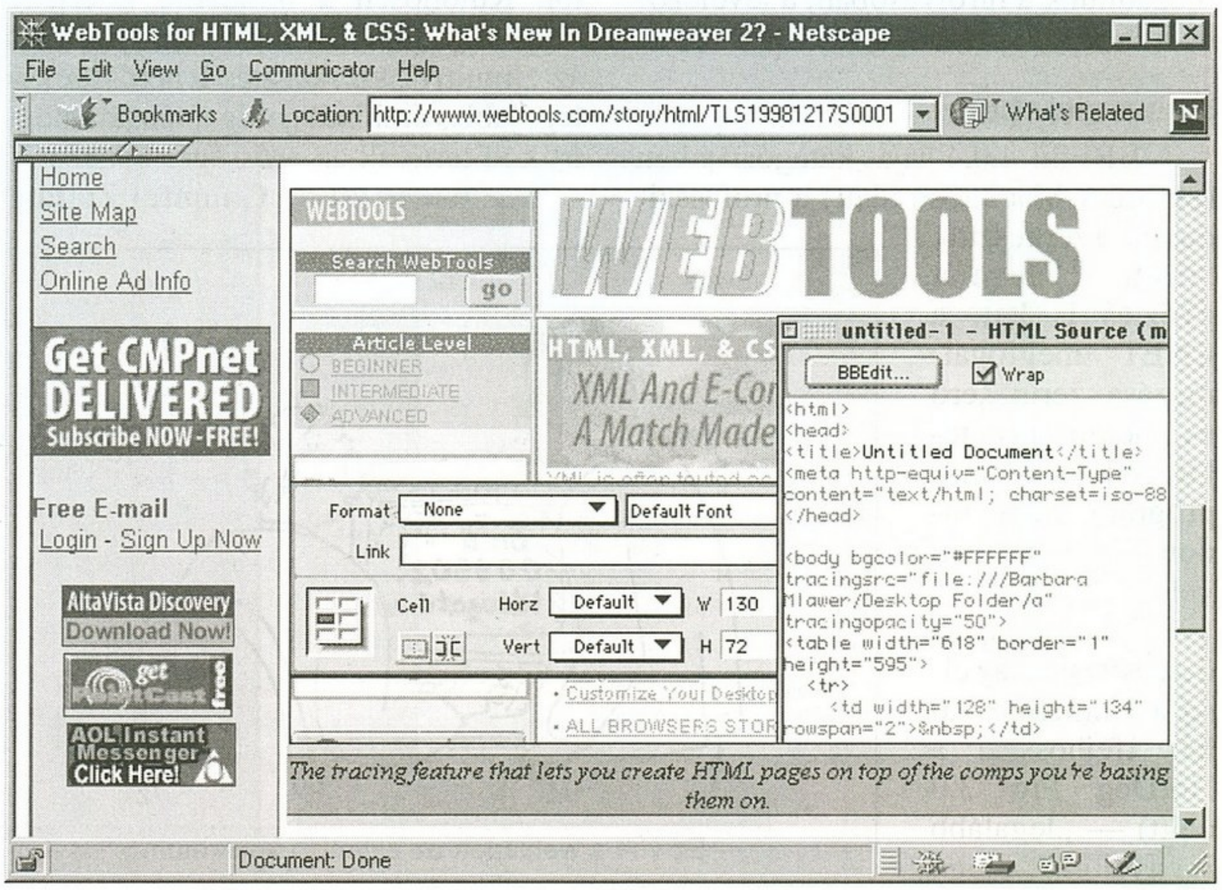
Értéklelési szempontok

A CNET 1988 végén az alábbi szempontok alapján értékelte a professzionális, tehát kereskedelmi forgalomban kapható, de azért kipróbálásra ingyenesen letölthető és (a BBEdit kivételével) Windows 9x alatt futtatható szoftvereket.

— *Felület.* Kényelmesen lehet-e kezelni a programot, könnyen megtalálják-e a számunkra fontos szolgáltatásokat, mennyire részletes és jól használható-e a helprendszer stb. Ebből a szempontból a HotDog Pro 3.0 bizonyult a legjobbnak.

— *Rugalmasság.* Ha az ember sokat és rendszeresen dolgozik egy szoftverrel, egyáltalán nem mindegy, hogy makrózással, billentyűkombinációk beállításával mennyire tudja azt saját igényei szerint „keze alá" formálni. Ezt leginkább a Webber 32 készítőinek sikerült megvalósítaniuk.

— *Validation.* Helyesség, azaz „helyesírás", a mindenkori HTML szintaxisának lehető legpontosabb követése és ellenőrzése. Ebből a szempontból is a Webber 32 bizonyult a legjobbnak, amely még azt is lehetővé teszi, hogy



Program	Rugalmasság	Kezelhetőség
Adobe PageMill 3.0	3	8
Claris Home Page 3.0	4	7
Elemental Drumbeat 2.0	8	6
GoLive Cyberstudio 3.0	7	8
Macromedia Dreamweaver 1.2	8	8
Microsoft FrontPage 98	6	8
NetObject Fusion 3.0	7	9
SoftQuad HotMetal Pro 4.0	7	6
Symantec Visual Page 2.0	5	7

ne csak az éppen aktuális HTML-verzió szintaxisa alapján vizsgáljuk meg weblapunkat, hanem valamelyik előző változat szerint is.

— *HTML támogatás.* A HTML-ben rejlő lehetőségek minél teljesebb kihasználása, a nem hivatalos, browserspecifikus címkék használatát is beleértve. Ebben a WebEdit Pro 2.0 utasította maga mögé a többieket: az ActiveX-ig és a Java appletekig mindenre lehetőséget nyújt.

— *Publikációs képességek.* Mennyire lehet jól dolgozni az editorral a Webre történő sokoldalú beillesztés kialakítása érdekében? A legjobb a Macintosh alatt működő BBEdit 4.0.4, amely nem csupán azt teszi lehetővé, hogy weblapjainkat „eftépézzük", hanem azt is, hogy azokat közvetlenül a szerveren szerkesszük. Hatékonyságát növeli, hogy együttműködik (a PC-s világban sajnálatos módon ismeretlen) freeware InternetConfiggal: ez gondoskodik róla, hogy e-mail címünket ne kelljen külön megadnunk a hírolvasóban, a levelezőben és az eftépézéshez.

Összesítésben

A BBEdit 4.0.4 saját kategóriájában (mármint Macintosh alatt) a professzionális szerkesztők között abszolút győztes is lett: a CNET megfogalmazása szerint „erő és flexibilitás tekintetében" más program nem tudott versenyezni vele.

PC-n viszont az összesítésben az első a HomeSite 4.0 lett (fejhosszal a HotDog Pro 3.0 előtt) — „legalább is egyelőre", ahogy az óvatos megfogalmazás mondja. Aki tehát elmélyültebb webszerkesztésre törekszik, valószínűleg érdemes ezek egyikével kezdenie.

A könnyebb utat keresők számára ott vannak a WYSIWYG editorok. A CNET velük kapcsolatban a rugalmasságot és a könnyű használhatóságot tekintette a legfontosabbnak. A fenti táblázat az erre vonatkozó értékelést összegezi, 10-es skálán osztályozva.

Mindent egybevetve a Dreamweaver találtatott a legjobbnak. A tesztelők külön kiemelték a beépített kódeditort (amely egyetlen egérkattintással elérhető), a DHTML híresen jó támogatását, valamint a „backward kompabilitást", vagyis azt, hogy ezzel a szoftverrel korábbi HTML verzióknak megfelelően is dolgozhatunk, amennyiben erre szükség van.

A másik, különösen ajánlott webszerkesztő a Fusion, még akkor is, ha ez inkább „desktop publisher", azaz kiadványszerkesztő, semmint HTML-editor. Különösen a mindenre kiterjedő drag and drop (és ennek megfelelően az „intuitív" szerkesztés lehetősége), a legújabb verziónál pedig a tisztán megírt kód tartozik az erősségei közé.

Galántai Zoltán



— Ez volt a weblapon az egyetlen konkrétum.

A hipertext karrierje

Tim Berners-Lee érdemei vitathatatlanok ugyan, de az is tény, hogy a HTML már egy hosszabb folyamat betetőzése volt. Az alábbiakan a hipertext előtörténetének legfontosabb állomásait ismertetjük.

■ 1945-ben az Atlantic Monthly hasábjain Vannevar Bush „As We May Think" címmel közzéteszi 1932-33 óta dédelgetett elképzeléseit egy mikrofilmen alapuló proto-hipertextes rendszerről.

■ 1965: Ted Nelson bevezeti a „hypertext" kifejezést, és egy olyan rendszert tervez, amely minden leírt szöveget magában foglal. Ez lett volna a Xanadu — és persze nem valósult meg.

■ 1967: A Brown University-n Andries van Dam vezetésével és az IBM támogatásával elkészül az első működő hipertext rendszer, a Hypertext Editing System, mely egy IBM/360-as mainframe-en fut. Később a Houston-i űrközpontban ilyen rendszerben készítik el az Apollo-program dokumentációját.

■ 1978: Aspen Movie Map, az első hipermédia-rendszer. Andrew Lippman és kollégiái fejlesztik ki az MIT Architecture Machine Group-nál (a mai Media Lab elődjénél). A számítógép képernyőjén keresztül, gyakorlatilag tetszőlegesen megválasztott útvonalon lehet virtuális túrát tenni, óránként 170 kilométeres sebességgel száguldva a Colorado-i Aspen városában.

■ 1985: Symbolics Document Examiner. Ez a felhasználók szélesebb körének szánt első hipertext rendszer: a Symbolcs munkaállomás 8000 oldalas kézikönyvét helyettesíti. Egy vizsgálat során 24 emberből csak ketten részesítették előnyben a nyomtatott változatot.

■ 1986: Az Office Workstation Limited (OWL) kiadásában megjelenik a Guide, a kereskedelmi forgalomban is kapható első hipertext rendszer. Előbb Macintosh-on, majd PC-n is fut.

■ 1987: A Macintosh minden gépéhez mellékeli saját hipertext rendszerét, a HyperCardot, és ezzel a hipertext a kuriózumból mindennapi használati eszközzé válik. Ugyanebben az évben megredezik az első hipertext tárgyú konferenciát is.

Folyton formálódó HTML nyelv

X-edik szintű találkozások

A HTML, a Web „közvetítőnyelve” eddig öt lépcsőn át jutott el a jelenleg érvényes 4.0 verzióig. Az előzményeknek, az előző szinteknek a megismerése egyfelől azért érdekes, mert segít történetileg követni, hogyan jött létre és vált egyre bonyolultabbá a hipertext leírónyelv; másfelől pedig azért, mert ez a fejlődési folyamat nagymértékben meghatározza a jelent és a közeljövőt — a Weben.

A HTML (Hypertext Markup Language) 1990 óta szolgál a honlapok leírására a Weben. Létrejött egy koordináló szervezet is, a World Wide Web Consortium (W3C, lásd az erről szóló cikket a 8. oldalon), amelyben a tagok többségének egyetértése szükséges minden új HTML elem bevezetéséhez, és egy új HTML verzióból csak akkor lesz W3C ajánlás, ha azt a többség elfogadja. Az eddigi tapasztalatok alapján a dolog működik: az ajánlásokat komolyan veszik, így a HTML jövője biztosnak látszik. (Vagy talán mégsem? A 3.2-es változatban már van egy időzített bomba, az APPLET címke. Ritka ugyanis az olyan Java applet, amely egyaránt működik Netscape Navigator és Internet Explorer alatt. Ha ilyenkor kapunk egy hibaüzenetet, akkor még olcsón megúsztuk. Előfordul azonban, hogy az applet egyszerűen „lelövi” a böngészőt, illetve az is, hogy sorozatos hibaüzenetek miatt a honlapot nem tudjuk megnézni.)

A HTML eddigi öt állomásához (Level 0, 1, 2, 3, 4) bizonyos kitérők is tartoznak, mint például a HTML 3.0 (amelyet végül is elvetettek, de a későbbi verziókban sok elemének hasznát vették), vagy a Netscape és a Microsoft által önhatalmúlag bevezetett újdonságok. Az alábbiakban megpróbálom összefoglalni, hogy az egyes szinteken milyen elemek jelentek meg, melyek váltak elavulttá, és általában milyen irányban haladt vagy halad a HTML. (Nem célom az egyes szintek minden elemének felsorolása, csak azokat említem meg, amelyek valamilyen szempontból fontosak. A részletezés a kézikönyvekben megtalálható.)

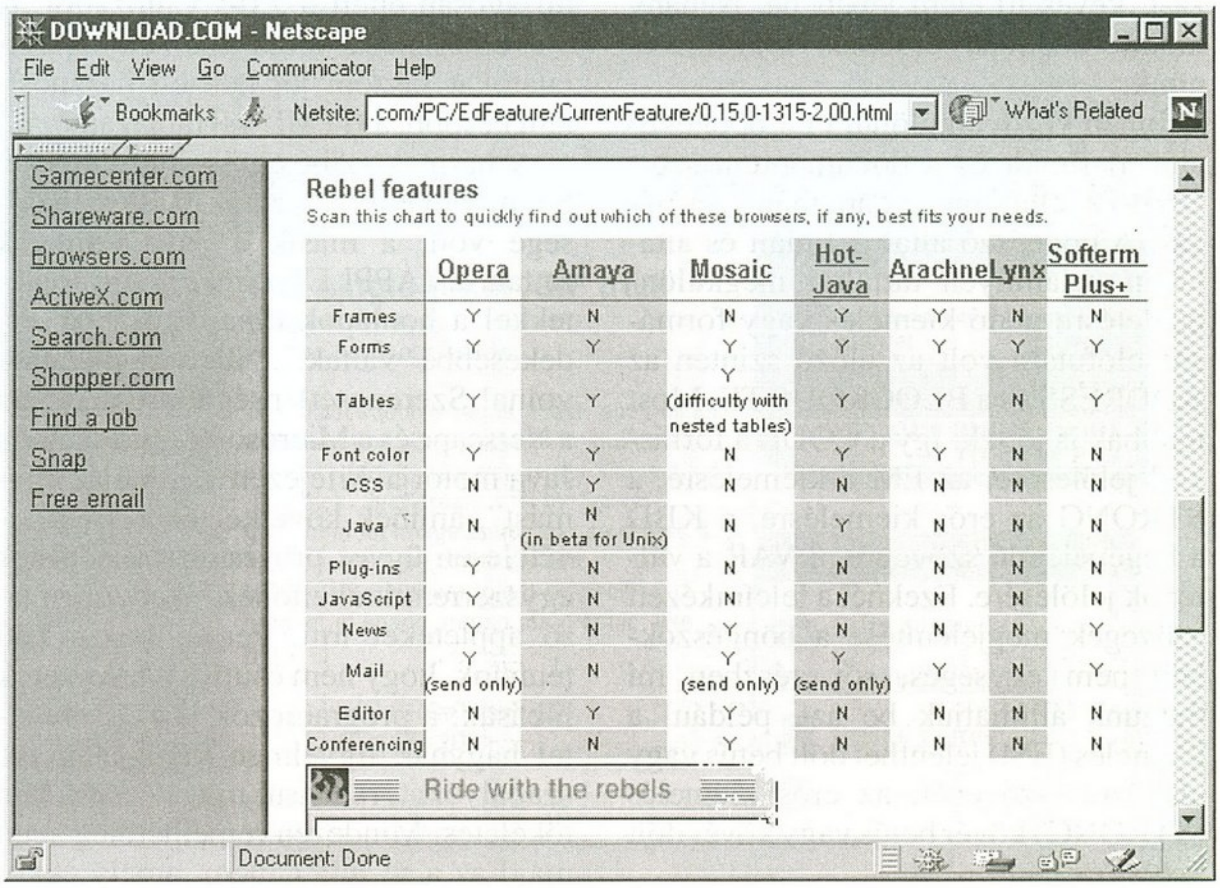
A nulladik szint

A nulladik szinten (Level 0) alakult ki a HTML dokumentumok alapvető szerkezete. Minden dokumentum a nyitó- és zárócímkén (HTML) belül elhelyezkedő fej (HEAD) és törzs (BODY) részből áll. Ezeknek az elemeknek egyike sem volt elhagyható, és ez a szabály a későbbi szinteken sem változott. Már ekkor is lehetőség nyílt viszont megjegyzések közbeiktatására, illetve a dokumentum típusának (szintjének) jelölésére, akár a HTML kezdőcímke előtt. Ez az úgynevezett Document Type Declaration (DTD), azaz dokumentumtípus-deklaráció, amely a későbbi szinteken különösen fontossá vált.

A címkék nyitórésze a címkenév, zárórésze pedig a címkenév megismétlése, előtte egy törtvonallal. Zárórész azonban nem minden címke esetében szükséges. A címkeneveket a HTML-dokumentumokban mindig csúcsos zárójelek közé kell tenni, és bár azokat e cikkben a könnyebb olvashatóság kedvéért elhagyjuk, a szintaxishoz természetesen mindig hozzátartoznak, tehát például <HEAD> és </HEAD>. A címkenevek kisbetűsen és nagybetűsen egyaránt írhatók, de a kiemelés kedvéért a cikkben a nagybetűs írásmódot választottuk.

A fejrészen belül kötelezően előírták a TITLE elem használatát, amelyben röviden megjelölhető a dokumentum tartalma. Lehetőség volt arra is, hogy a dokumentumnak más dokumentumokhoz és objektumokhoz való viszonyát a LINK elem segítségével megadjuk, míg a META elem révén a dokumentummal kapcsolatos egyéb információk kaphattak helyet (mint például a szerző neve). Az ISINDEX elemmel egyszerű kereséseket lehetett végrehajtani.

A tényleges tartalom a törzsben (BODY) helyezkedett el, ide került az újabb HTML változatokban is meglévő összes alapvető elem, mint a dokumentumok összefűzésére (linkelésére) használatos horgony (az anchor szóból: A); a többféle rendezett és rendezetlen lista (OL, UL, DIR, MENU); a definíciólista (DL); az idézet (BLOCKQUOTE); az előre formázott szöveg (PRE) és a tulajdonosi jogok, illetve a szerző jelölésére alkalmas elem (ADDRESS). A dokumentum formázására szolgáltak a címsorok (H1...H6); az elválasztók (például a vízszintes vonal: HR); az új bekezdés (P) vagy a sortörés (BR). Képek beillesztésével is változatosabbá



	Opera	Amaya	Mosaic	Hot-Java	Arachne	Lynx	Softerm Plus+
Frames	Y	N	N	Y	Y	N	N
Forms	Y	Y	Y	Y	Y	Y	Y
Tables	Y	Y	Y (difficulty with nested tables)	Y	Y	N	Y
Font color	Y	Y	N	Y	Y	N	N
CSS	N	Y	N	N	N	N	N
Java	N	N (in beta for Unix)	N	Y	N	N	N
Plug-ins	Y	N	N	N	N	N	N
JavaScript	Y	N	N	N	N	N	N
News	Y	N	Y	N	N	N	Y
Mail	Y (send only)	N	Y (send only)	Y (send only)	Y	N	Y
Editor	N	Y	Y	N	N	N	N
Conferencing	N	N	Y	N	N	N	N

tehettük honlapunkat (IMG elem), és megvolt a lehetőség „image map" (elhelyezési áttekintés) készítésére is (IMG elem ISMAP attribútummal). Ez utóbbi még nem volt önálló HTML elem, ezért meglehetősen boyolult módon kellett hozzá az „image map"-hez kapcsolódó linkeket tartalmazó fájl is, meg egy CGI, hogy az egész működjön. Később, a 3.2-es verzióban mindez leegyszerűsödött.

Már ezen a szinten megjelentek az ún. „karakterentitások", a több karakterből álló, valójában azonban egyetlen egységet (entitást), tehát jelet vagy speciális karaktert jelentő elemek, például a < jelölése (ha azt ki akarjuk íratni, és nem címkehatárként használjuk) < (az entitások nyitókaraktere az & jel, lezárója pedig a pontosvessző). A teljes Latin 1 karakterkészletet használhattuk, az ékezetes betűket entitások definiálásával írták le (például á = á).

Mindent egybevetve, már a nulladik szintű HTML alkalmas volt arra, hogy egyszerű honlapokat készítsünk vele: megjelentek benne az alapvető formázóelemek, és az entitások segítségével az ékezetes betűket használó nyelveken is el lehetett készíteni a dokumentumokat (bizonyos korlátok és kompromisszumok között, lásd szép magyar nyelvünk mostoha ő és ű betűit, melyek ékezete csak ^ és ~ formában jeleníthető meg). Mindez arra volt elegendő, hogy egy jól elkészített weblap „pofára" hasonlítson egy egyszerű nyomtatott magazinhoz.

Egyes szint

A következő szint (Level 1) az előzőhöz képest nem jelentett nagy változást, kevés új elem lépett be. Néhány dolgot azonban érdemes megemlítenünk.

Innen kezdve különül el a böngésző által definiált és a dokumentum szerkesztője által meghatározható formázás. A böngésző által definiált és általában valamilyen tartalmi megkülönböztetésre utaló kiemelés vagy formázás előfutára volt az előző szinten az ADDRESS és a BLOCKQUOTE. Most újabbak is jöttek: így a CODE a forráskód jelölésére, az EM a kiemelésre, a STRONG az erős kiemelésre, a KBD a begépelendő szövegre, a VAR a változók jelölésére. Ezeknél a felcímkézett szövegek megjelenítése a böngészőkben nem egységes, sőt, részben mi magunk állíthatjuk be azt, például a kiemelés (EM) jelenthet dőlt betűs vagy aláhúzott szöveget, az erős kiemelés (STRONG) kövér betűs vagy kövér dőlt betűs írást stb.

A dokumentum készítője által egyértelműen meghatározható karakterformázásra három lehetőség volt: a B (kövér), az I (dőlt) és a TT (írógép típusú, vagyis azonos szélességű betűk). Ezekkel a címkékkel lehet utasítani a böngészőt, hogy a kijelölt szövegrészt az éppen érvényes betűtípus kövér vagy dőlt változatával jelenítse meg, illetve az ott definiált írógép típusú betűkészletre váltson át. (Ez utóbbi kicsit kakukktojás, mert ilyen betűtípus többféle van, de lényegüket tekintve persze azonosak.) Az egyes szint tehát nem hozott sok újdonságot. Ez volt a HTML 2.0-s változata, pontosabban a 2.0-s változat első szintje.

Kettes szint

A kettes szinten (Level 2) a korábbiakhoz képest jelentős változás az űrlap (FORM) bevezetése. Előzőleg az ISINDEX elem már nyújtott bizonyos korlátozott lehetőséget az adatközlésre, de a FORM sokkal kifinomultabb eszköz, és előrevetítette az ISINDEX későbbi megszűnését is. Az űrlapok révén lehetőség nyílt a honlapokon többsoros szövegek bevitelére (TEXTAREA), választási lehetőségek felkínálására (SELECT, OPTION), illetve egyéb módon történő információgyűjtésre (INPUT elem, például egysoros szöveg bevitelére). A HTML 2.0 változatának a második szintje szabvány lett, ez szolgált kiindulási alapul a további HTML változatokhoz.

Hármas szint

A HTML 3.2 verziót (Level 3) 1996-ban fogadta el és véglegesítette a W3C. (A 3.0-sból soha nem lett ajánlás, mert túlságosan eltért a 2.0-s változattól. A 3.0 elkészítése során szerzett tapasztalatokat viszont jól lehetett hasznosítani később, a HTML 4.0 táblázatainál.)

A hármas szint egyik legfontosabb bővítménye a táblázatok beilleszthetősége volt, a másik a Java appletek futtatása (APPLET címke). Az appletekkel a honlapok dinamikusabbá, érdekesebbé váltak... illetve válhattak volna! Szerencsétlenségünkre azonban a Netscape és a Microsoft böngészőinek Java motorjai kifejezetten „utálják egymást", aminek következtében csak kivételesen ügyes programozók képesek egyszerre mindkettőnek a kedvében járó appleteket írni. Persze hozzá kell tennünk, hogy nem csupán a browserek hibásak: a programozók is előszeretettel hagynak figyelmen kívül alapvető szabályokat, ráadásul maga a Java sem tökéletes. Mindezen remélhetőleg segít majd az a SCRIPT elem, amely ekkor jelent meg, de jelentős szerephez csak később jutott. A SCRIPT elemnek köszönhetően a HTML-dokumentumokba nemcsak Java appletek, hanem különböző scriptek is beilleszthetők. A HTML 4.0 számára a SCRIPT-hez hasonlóan előre lefoglalták a STYLE címkét is, ezzel mintegy megelőlegezve a stíluslapok megjelenését.

Néhány további bővítés: sok elemhez új attribútumok váltak illeszthetővé, például rendezett lista esetén a felsorolás betűvel és számmal is jelölhető, módosíthatóvá vált a háttérszín stb. Új formázóelemek kerültek a nyelvbe, amilyen például a FONT a karakterek színének és méretének módosítására; vagy a DIV, amellyel bekezdéseket definiálhatunk, és jobbra, balra vagy középre igazíthatunk. Egyszerűbbé vált az image map készítése a MAP elem segítségével.

A HTML 3.2 két legfontosabb újítása mindenképpen a táblázatok és az appletek beillesztésének lehetősége, a többi módosítás inkább még csak ígéret volt (STYLE, SCRIPT), vagy az eddigi lehetőségek kibővítése. Sikerült megőrizni a 2.0-s változathoz felülről lefelé irányuló kompatibilitást (nem úgy, mint a 3.0 verziónál), és később jó kiindulópontként szolgált a HTML 4.0-hoz is. Az új formázóelemeknek köszönhetően színesebbé, összetettebbé válhattak a honlapok (a Java is ezt a cél szolgálja, de vele egyelőre sajnos óvatosan kell bánni). Hozzá kell persze tenni, hogy az új lehetőségek ahhoz is táptalajul szolgáltak, hogy sok honlap készítője ízléstelen összevisszasággá, kaotikussá formálhassa alkotását.

Négyes szint

A négyes szint pillanatnyilag a legfrissebb. Ez az 1997 nyarán kibocsátott 4.0-s verzió. Minden bizonnyal az eddigi legnagyobb ugrás a HTML történetében, mert a korábbi szintekhez képest alapvető változtatásokat hajtottak végre, és számos új elemet vezettek be.

A HTML 4.0 megalkotásánál teljesen új szempontokat is figyelembe vettek. Igyekeztek például a vakok és gyengénlátók hozzáférési lehetőségeit is kiszélesíteni a beszédszintetizátor és a Braille-olvasó támogatásával.

Megpróbálták ellensúlyozni az angol nyelv dominanciáját is. Többféle karakterkészlet kezelésével könnyebbé vált más nyelvű honlapok készítése, lehetőség nyílt akár jobbról balra olvasandó szövegek megjelenítésére is (amilyen az arab vagy a héber írás).

Megoldották, hogy a Weben található információk többféleképpen elérhetők

legyenek, és hogy a honlapokat az ehhez szükséges képességekkel fel lehessen ruházni. Ide tartozik a mobil telefonon keresztüli hozzáférhetőség vagy a beszédszintetizátor, a kis sávszélességű csatornák és a csak szöveges böngészők használata stb.

Továbbfejlesztettek néhányat a már meglévő elemek közül is, például a táblázatokat, az űrlapokat. Legalizálták a frame-ek (keretek) alkalmazását. Ezek a sok vitára okot adó elemek addig „hivatalosan" nem tartoztak a HTML nyelvhez, de a Netscape Navigator és az Internet Explorer révén polgárjogot nyertek, a weblapkészítők pedig előszeretettel, bár korántsem elég körültekintően használták azokat. A pontos specifikációval remélhetőleg már csökken a frame-ek körüli zűrzavar.

A böngésző által definiált formázóelemek is tovább gyarapodtak (INS, ACRONYM). Néhánynak a helyére újak léptek: ISINDEX helyett FORM, APPLET helyett OBJECT, CENTER helyett DIV.

Az új HTML-nek fontos része a multimédia támogatása. Az első ilyen elem az egyszerű állókép volt (IMG címkével definiálva), most az OBJECT elem segítségével bármilyen multimédiás objektumot (képet, videót, hangot, matematikai kifejezést, appletet) beilleszthetünk a dokumentumba. A Java-krízis enyhítésére hivatott SCRIPT elemről az előbbiekben már szóltunk, de a teljesség kedvéért hozzá kell még tennünk, hogy a szkeptikusok szerint ezzel ismét egy szellemet szabadítottak ki a palackból, amelyen utána már nem lehet úrrá lenni.

Stíluslapok

A korábbiakhoz képest a legnagyobb ugrást a stíluslapok bevezetése hozta. Ezekkel egyszerűbbé, gyorsabbá vált a dokumentumok formázása, tehát a korábban formázóelemekkel megkülönböztetett szövegrészeket is célszerűbb stíluslapokkal definiálni. De ez nem jelenti azt, hogy teljesen helyettesíthetik a dokumentum szerkezetét leíró elemeket. Például a címsorok kiemelése (H1...H6 elemek) stíluslappal is könnyen létrehozható, az mégsem ugyanaz, ha egyébként ugyanúgy néz is ki. A böngésző által definiált formázóelemeket nem tanácsos stíluslappal helyettesíteni, mert azok a dokumentum struktúrájához tartoznak. A másik fontos alapelv az, hogy a honlapoknak stíluslapok nélkül is működőképeseknek kell maradniuk. Persze ez esetben nem fognak ugyanúgy tündökölni, de a szerkezet és az olvashatóság fontosabb.

A tiltások és megszorítások után most vegyük szemügyre közelebbről, hogyan néznek ki a stíluslapok.

A stíluslapok leírására a CSS (Cascading Style Sheets) nyelv szolgál. Egy stíluslap valójában a HTML-dokumentumokhoz kapcsolható sablon, amely meghatározza a megjelenési formát (a színeket, a karaktertípust stb.). Többféle elhelyezési módja is van:

— Alkalmazható ún. külső stíluslapként (external style sheet), amikor külön fájl. Kiterjesztése kötelezően .CSS, és a stílusra vonatkozó információkat CSS nyelven tartalmazza. A HEAD részen belül elhelyezett LINK elemmel csatolható bármely HTML-dokumentumhoz, egyszerre többhöz is.

— A másik alkalmazási mód a beágyazott stíluslap (embedded style). Ilyenkor csak egy meghatározott dokumentumra vonatkozik, és nem külön fájlban, hanem a dokumentum STYLE elemeként kell megadni, szintén CSS nyelven, mégpedig egy megjegyzés (comment) címkén belül, hogy ne zavarja meg a stíluslapok értelmezésére nem képes bögészőket.

— A STYLE segítségével meghatározható egyetlen elem stílusa is (inline style), ilyenkor azt közvetlenül az elem kezdőcímkéje után kell elhelyezni.

— A legérdekesebb lehetőségnek a behívható stíluslapok használata ígérkezik (imported style). Egy honlaphoz így több stíluslapot csatolhatunk, melyek közül az egyik a dokumentum egészére vonatkozó fő stíluslap, azt egészítik ki a mellékstíluslapok, csak a dokumentum bizonyos részeinek formáját határozva meg. A tervek szerint a jövőben megtehetjük majd azt is, hogy alternatív fő stíluslapokat adunk meg, és ezek közül a választást a felhasználóra bízzuk.

A fejlesztők gondoltak a régebbi, a HTML 4.0-s változatát értelmezni képtelen böngészők használóira is. A dokumentum elején a dokumentumtípus megadásánál három lehetőség közül választhatunk. Ezek:

— HTML 4.0 Transitional. Áthidaló változat, amely már felhasználja az új lehetőségeket, de használható a régebbi böngészőkkel is.

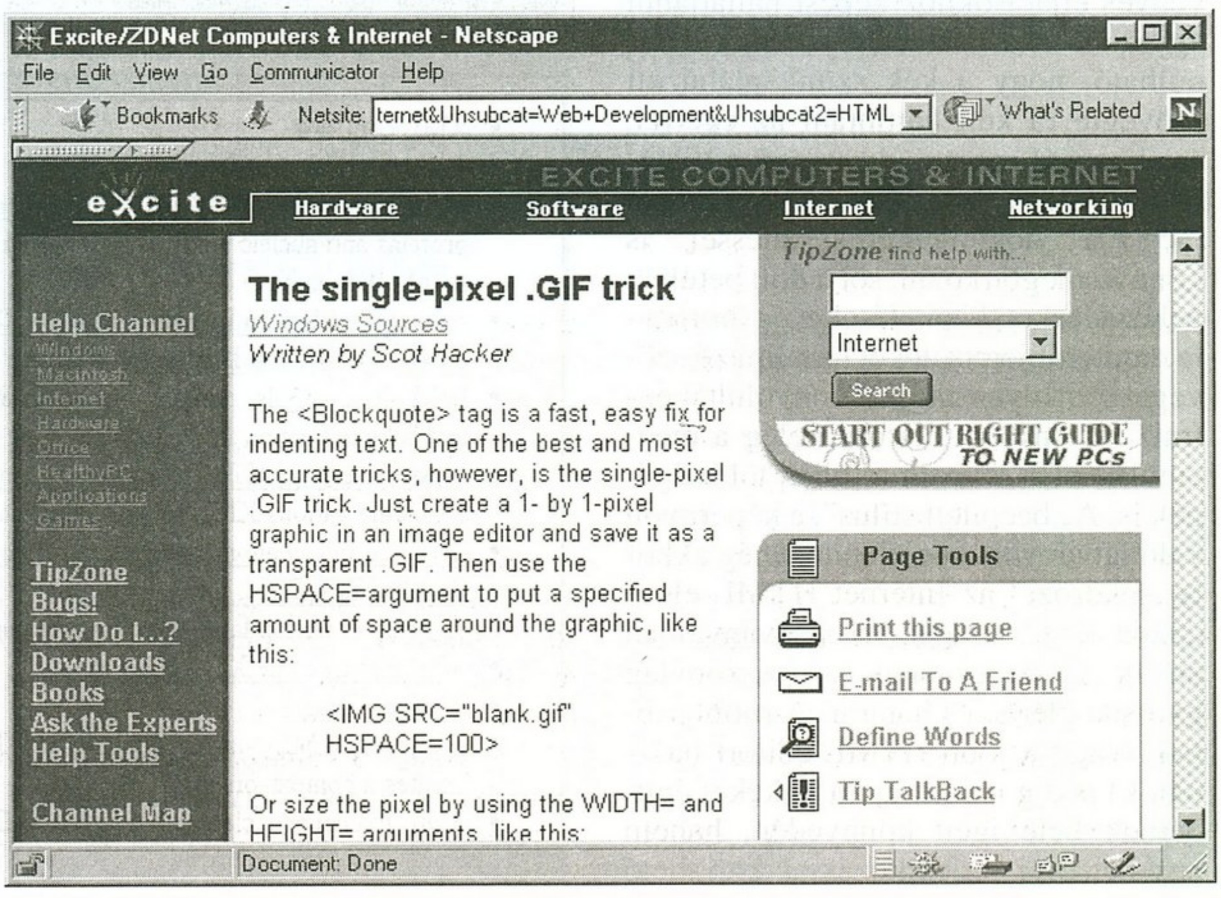
— HTML 4.0 Strict. Szigorúan csak stíluslapokat használ, mellőzi a karakterformázó címkéket.

— HTML 4.0 Frameset. Elnevezése értelemszerűen a frame-ek alkalmazására utal.

Pillanatnyilag tehát itt tartunk. A fejlődés trendje olyan dinamikus honlapok felé mutat, amelyek valós időben, böngészés közben képesek átalakulni, a körülményekhez alkalmazkodni, a felhasználókkal kommunikálni. Ezeket a feladatokat oldhatják majd meg a Java appletek és az újonnan bevezetett scriptek.

Valószínűleg egyre nagyobb szerephez jut a multimédia, bár azt a hálózat jelenlegi áteresztőképessége erősen korlátozza. A filmek és a zeneszámok lejátszása a sáv jelentős részét lefoglalná, a többi felhasználót pedig a guta kerülgetné, ha a remélt gyorsulás helyett tovább lassulna a kommunikáció. Tekintve, hogy minden kapacitás nagyon gyorsan kimerül, ha az erőforrásigényes alkalmazások tömegessé válnak, a multimédia elburjánzásának ez egyelőre határt szab.

Mákos András



Félúton a HTML és az SGML között

Irány az XML

Manapság, amikor egyes felmérések szerint a fejlett országok nemzeti jövedelmüknek akár 20%-át is információk előállítására (és persze megjelenítésére) fordítják, ezen információk 90%-a még mindig nem adatbázisokban, hanem dokumentumokban kerül rögzítésre. Elképzelhető tehát, hogy mennyi munkát (és persze mekkora költséget) jelent a hagyományos szövegszerkesztőkkel megírt anyagok „webképessé" tétele, HTML-be való konvertálása. Ezen a helyzeten kíván segíteni az XML, vagyis az Extensible Markup Language, a kiterjesztett jelölőnyelv, amely a szakértők szerint hamarosan a Web teljes átalakulását fogja eredményezni.

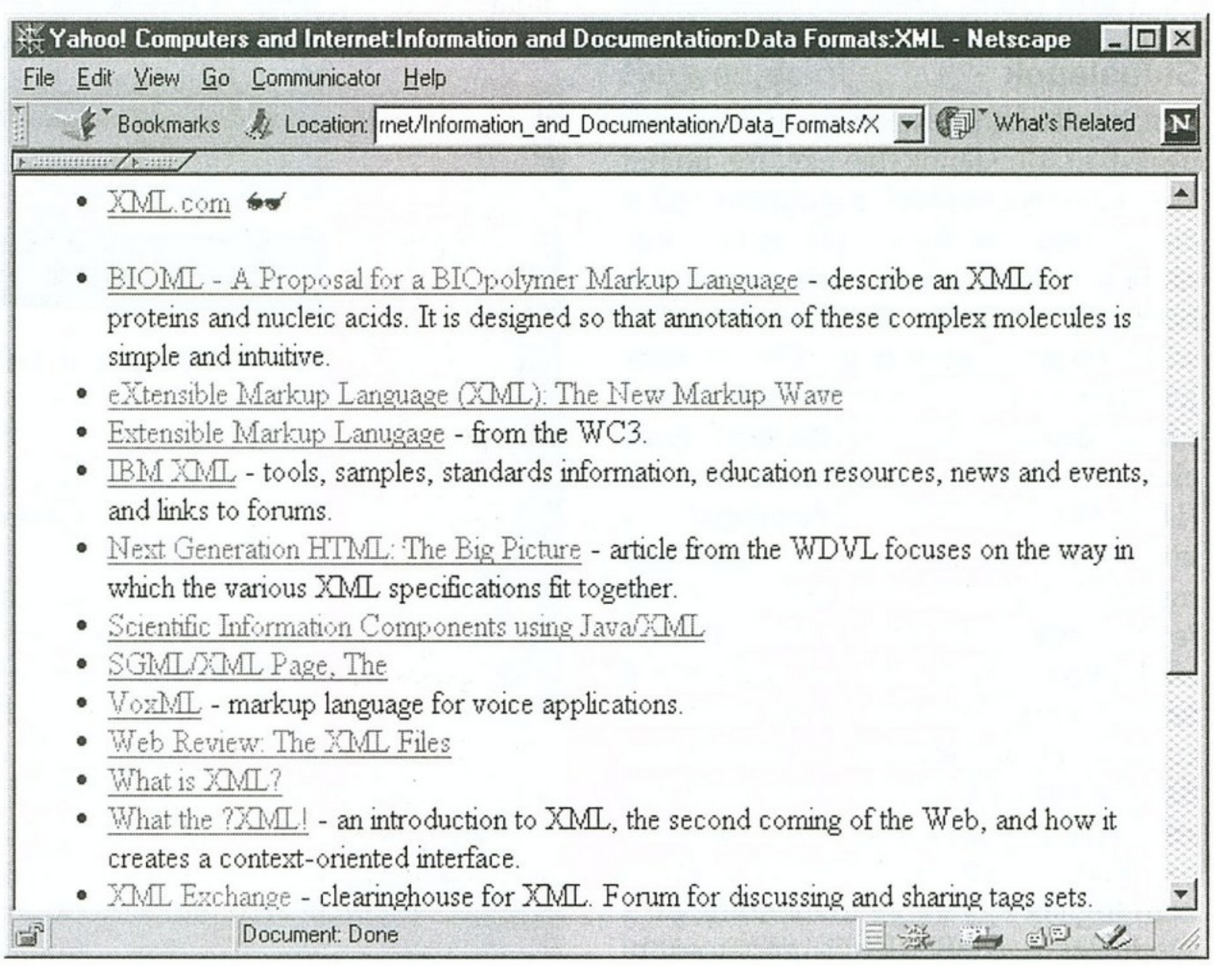
Teljesen indokolt, hogy elgondolkodjunk a HTML problémáit megoldani hivatott XML-ről, de azért alapos oka van annak is, hogy a Tim Berners-Lee és csapata által létrehozott HTML egyelőre olyan nagy népszerűségnek örvend. Emberek milliói, sőt tízmilliói írták és írják meg saját weblapjaikat a HTML-t használva, és mintha az egységes szövegformátum szerepét is ez kezdené betölteni. Egyes műszaki egyetemeken ma már elfogadják, ha a hallgató nem a WinWord legújabb erőforrás-zabáló verziójával adja be dolgozatát (lehetőleg makrovírussal együtt), hanem HTML-ben megformázva.

Az egyszerűség ára

A HTML előnyei vitathatatlanok. A számítástechnikában megszokott tekervényes eljárásokhoz képest hallatlanul egyszerű a használata. Gyorsan megtanulható, hogy a kék színű, aláhúzott szövegre rá kell kattintani az egérrel, és akkor jól-rosszul, de az egész Világhálón boldogulni fogunk. Alapszinten még a dokumentumkészítéssel is könnyű megbirkózni, sőt a dőlt betűket, bekezdéseket, kiemeléseket és (horribile dictu) lábjegyzeteket tartalmazó szöveget sem olyan nagyon bonyolult megírni. Néhány óra biztosan elég a megtanulására a 25 éven felüliek többségének is. A „beépített stílus", a képernyőn való látványos megjelenítés még akkor is „lekörözi" az Internet HTML előtti „plain text" világát, ha nyilvánvalóan ennek is megvannak a (viszonylag gyorsan elérhető) határai. A rutinosabbak (vagy a jobb HTML-editort használók) pedig nem csupán linkeket építhetnek bele igen könnyedén, hanem egy-két órai próbálkozással kitölthető űrlapokat is készíthetnek vagy scripteket illeszthetnek be a megfelelő helyekre.

Ennek az egyszerűségnek és könnyen kezelhetőségnek is megvan azonban a hátulütője: a limitáltság, a korlátozottság. Vagyis sok mindent meg lehet vele csinálni, de korántsem mindazt, amire technikai lehetőség lenne. Esztétikai oldalról nézve pedig túlságosan statikus az egész. Mivel fix és módosíthatatlan elemekből épül fel, ezek segítségével rögzített, rugalmatlan struktúrák jönnek létre, ami komoly nehézségekhez vezethet. Becslések szerint csak az Amerikai Egyesült Államokban évente legalább 100 milliárd (!) dokumentumot állítanak elő, és ha egy cég árjegyzékeit, termékeinek teszteredményeit vagy bármi mást papíron és Interneten is hozzáférhetővé akarja tenni, akkor az nem kevés pluszmunkát fog eredményezni. A konvertálgatás időtrabló és költséges (különösen ilyen mennyiségben), és bár bizonyos szövegszerkesztők (lévén önmagukkal sem kompatibilisak) ehhez már hozzászoktattak bennünket, a HTML esetében számolni kell azzal is, hogy rendszerint egyedileg, „manuálisan" szükséges befejezni a formázásokat és a kiigazításokat.

További komoly gond, hogy a címkék (tag) lényegében csak arra szolgálnak, hogy a dokumentumok megjelenését írják le, az pedig nem nagyon teszi lehetővé az adatcserét. Egyszerű példával illusztrálva, képzeljük el, hogy egy ipari termékre vonatkozó adatokat (ellenállás, feszültség, hőtágulás stb.) szeretnénk feltenni a Webre. Mivel azonban a HTML-ben nincsenek ilyen speciális elemek, vagyis nem tudjuk ugyanúgy beírni a dokumentumba, hogy <ellenállás>...</ellenállás>, ahogyan mondjuk a kövér szedést: ..., ezért kénytelenek vagyunk lebutítani az egészet, és aki az adatokat a Webről letölti, hasonlóképpen kény-



telen lesz azt kézzel áttenni a megfelelő adatbázisokba, mert nincs olyan szoftver, amely értelmezni tudná a HTML-ben szövegként jelen lévő információkat. Pedig ha a HTML nem lenne ennyire kötött, az ilyesmi egyáltalán nem volna lehetetlen. Egy webböngésző is „megérti" az adott oldal formázására vonatkozó elemeket, és elvileg semmi akadálya nincs annak, hogy egy leírónyelv akár diódaméretezési adatok, akár Shakespeare műveinek elemzésére szolgáljon (lásd a mellékelt keretes anyagot). Na persze, nem a HTML.

Általános jelölési elvek

Lényegében ugyanezzel a problémával szembesülünk akkor is, amikor egy keresőrendszerbe beírunk egy szót, és kezelhetetlenül sok találatot kapunk. Ha elérhetnénk, hogy egy Zeus márkájú hifitorony után kutatva a szoftver a görög istenekkel, a mitológiával és hasonlókkal foglalkozó online forrásokat figyelmen kívül hagyja, akkor persze könnyebb dolgunk lenne. Logikai szűrőfeltételekkel operálva jelentősen szűkíteni tudjuk a kört, de néha a megfelelő kritériumok felállításának kigondolása több időbe telik, mint végignézni a buta keresés eredményéből az első 30 találatot, melyek között nagy valószínűséggel ott szokott lenni az is, amit keresünk. Elvileg a Webet felépítő hipertext rendszer is a segítségünkre lehetne a keresésben, ha egy ilyen termék neve az <electronics>...</electronics> címkepár közé lenne beírva.

Nos, éppen az ilyen feladatok megoldására lett kifejlesztve a Standard Generalized Markup Language (SGML). A szabványként 1986-ban elfogadott univerzális leírónyelv pontosabb megértéséhez érdemes előbb tisztázni néhány fogalmat.

Ott van először is a WinWordtől a Quark XPressig a legtöbb szerkesztési szoftver által használt ún. procedural markup (eljárásjelölés). Ez a gyakorlatban azt jelenti, hogy mindegyik szoftver saját jelölőrendszerrel dolgozik (ezért kell az egyik szövegszerkesztővel létrehozott anyagot konvertálni, ha másik szoftverrel akarom azt tovább szerkeszteni), továbbá hogy a jelölés általában a formázókódokat definiálja, és hogy ezek össze vannak keverve magával a szöveggel. A következmény az, hogy amelyik megjelenítési forma megfelelő például a nyomtatáshoz, az változtatás nélkül nem alkalmas arra, hogy felkerüljön az Internetre.

A descriptive markup (leíró jelölés) vagy más néven generic markup (egységes jelölés) egészen másképp műkö-

```
<?xml version="1.0"?>
<!DOCTYPE PLAY SYSTEM "play.dtd">

<PLAY>
<TITLE>The Tragedy of Hamlet, Prince of Denmark</TITLE>

<FM>
<P>Text placed in the public domain by Moby Lexical Tools, 1992.</P>
<P>SGML markup by Jon Bosak, 1992-1994.</P>
<P>XML version by Jon Bosak, 1996-1998.</P>
<P>This work may be freely copied and distributed worldwide.</P>
</FM>

<PERSONAE>
<TITLE>Dramatis Personae</TITLE>

<PERSONA>CLAUDIUS, king of Denmark. </PERSONA>
<PERSONA>HAMLET, son to the late, and nephew to the present
king.</PERSONA>
<PERSONA>POLONIUS, lord chamberlain. </PERSONA>
<PERSONA>HORATIO, friend to Hamlet.</PERSONA>
<PERSONA>LAERTES, son to Polonius.</PERSONA>
<PERSONA>LUCIANUS, nephew to the king.</PERSONA>

<PGROUP>
<PERSONA>VOLTIMAND</PERSONA>
<PERSONA>CORNELIUS</PERSONA>
<PERSONA>ROSENCRANTZ</PERSONA>
<PERSONA>GUILDENSTERN</PERSONA>
<PERSONA>OSRIC</PERSONA>
<GRPDESCR>courtiers.</GRPDESCR>
</PGROUP>

<PERSONA>A Gentleman</PERSONA>
<PERSONA>A Priest. </PERSONA>

<PGROUP>
<PERSONA>MARCELLUS</PERSONA>
<PERSONA>BERNARDO</PERSONA>
<GRPDESCR>officers.</GRPDESCR>
</PGROUP>

<PERSONA>FRANCISCO, a soldier.</PERSONA>
<PERSONA>REYNALDO, servant to Polonius.</PERSONA>
<PERSONA>Players.</PERSONA>
<PERSONA>Two Clowns, grave-diggers.</PERSONA>
<PERSONA>FORTINBRAS, prince of Norway. </PERSONA>
<PERSONA>A Captain.</PERSONA>
<PERSONA>English Ambassadors. </PERSONA>
<PERSONA>GERTRUDE, queen of Denmark, and mother to Hamlet. </PERSONA>
<PERSONA>OPHELIA, daughter to Polonius.</PERSONA>
<PERSONA>Lords, Ladies, Officers, Soldiers, Sailors, Messengers, and
                  other Attendants.</PERSONA>
<PERSONA>Ghost of Hamlet's Father. </PERSONA>
</PERSONAE>

<SCNDESCR>SCENE Denmark.</SCNDESCR>

<PLAYSUBT>HAMLET</PLAYSUBT>

<ACT><TITLE>ACT I</TITLE>

<SCENE><TITLE>SCENE I. Elsinore. A platform before the castle.</TITLE>
<STAGEDIR>FRANCISCO at his post. Enter to him BERNARDO</STAGEDIR>

<SPEECH>
<SPEAKER>BERNARDO</SPEAKER>
<LINE>Who's there?</LINE>
</SPEECH>

<SPEECH>
<SPEAKER>FRANCISCO</SPEAKER>
<LINE>Nay, answer me: stand, and unfold yourself.</LINE>
</SPEECH>

<SPEECH>
<SPEAKER>BERNARDO</SPEAKER>
<LINE>Long live the king!</LINE>
</SPEECH>
```

dik, mert a dokumentumban a szöveg funkcióját határozza meg, nem pedig annak konkrét fizikai megjelenését. A szöveg és a formázás tehát elkülönül egymástól. „A descriptive markup — írja az SGML-lel foglalkozó egyik internetes forrás — a dokumentum struktúráján alapul, és ezen a struktúrán belül az egyes elemeket (amilyen a fejezet, a bekezdés vagy a tartalomjegyzék) olyan jelölési mód segítségével határozza meg, amely nem azt mondja el, hogy az adott elemnek milyen a konkrét formája, hanem hogy az micsoda." Az anyagot így akár kinyomtathatjuk, akár bármilyen más formában közzétehetjük. Az egységes jelölés előnyei a mai szövegszerkesztők által használt módszerekkel szemben teljesen nyilvánvalóak.

SGML, a csodaszer

Az SGML két okból is több ennél, illetve más. Egyfelől azért, mert mintegy általános kereteket ad: meghatározza a dokumentumba beépíthető descriptive markup formáját, másfelől pedig, és ez talán még fontosabb, meghatározza a dokumentum struktúrájának leírásakor használandó eljárást is. Ami azonban nem jelenti azt, hogy ne tudnánk akárhány jelölőnyelvet (markup language) felépíteni az SGML által meghatározott feltételrendszeren belül.

A HTML is az SGML-nek nagyon specializált és nagyon leszűkített változata. Kitalálhatnánk egészen fantasztikus és valószínűtlen jelölőnyelveket is, mondjuk olyat, amely a sumér agyagtáblák átírására szolgál. Az ezekkel készült dokumentumok az egyszeri megírás után nem csupán „platformfüggetlenek" lennének (vagyis akár a Weben, akár a nyomdában dolgozhatnánk velük), hanem „újrafelhasználhatóak" is.

Újrafelhasználáson a következőket kell érteni. A HTML fejlődése tulajdonképpen újabb címkék hozzáadásával (vagy kihullásával) jellemezhető, és ha valaki a 3.0-s változat alapján írta meg a weblapját, akkor a 3.2-es változat megjelenésekor nekiállhatott az egészet átírni, de még a legújabb, a 4.0-s is megváltoztatott jó néhány dolgot. Aki tehát a browserek jövendő generációi által is tökéletesen „látható" akar maradni, az mindig fogja magát, és átírja a megváltozott kódokat — ami egy nagy cég olykor sok száz oldalt tartalmazó webhelyén valóban sziszifuszi és idegőrlő munka lehet. Rendszergazdáktól hallható gúnyos megjegyzések szerint mostanában bizonyára azért nőtt hat hónapról egy évre a Netscape Navigator és az Internet Explorer közötti párharc újabb felvonásai közé iktatott szünet, mert ha gyakrabban jönnének ki a legújabb HTML-t „beszélő" böngészőkkel, akkor ők is (mármint a Microsoft és a Netscape) gyakrabban lennének kénytelenek újraírni azt a rengeteg weblapot.

Az SGML kényelmes megoldást kínál az ilyen problémákra. Ma már sok helyen SGML-ben tárolják a dokumentumokat, és az újabb HTML-verzió megjelenésekor a megfelelő szoftverekkel egyszerűen konvertálják azokat az új jelölési szabályoknak megfelelően. A más módon esetleg több hétig tartó munkát így néhány óra alatt el lehet végezni.

Ezek után már csak az a kérdés, hogy akkor miért fusizunk mégis HTML-ben, ha az SGML előnyei ennyire egyértelműek. A válasz az, hogy éppen ezért. Az SGML túl jó, illetve túl bonyolult ahhoz, hogy az átlagfelhasználó ezzel dolgozzon, ezért a browserek sem támogatják. Egy SGML-es levelezőlistára felkerült megfogalmazás szerint a Standard Generalized Markup Language olyan gyógyszer, amelytől a súlyos betegek meggyógyulnak, a kevésbé súlyosak viszont meghalnak.

Arany középút

Az SGML csak a struktúrával foglalkozik, és még akkor sem támogatja a stílusokat (style), ha létezik néhány „stylesheet standard" is: ott van például az eredetileg az amerikai hadsereg részére kidolgozott FOSI (Formatting Output Specification Instances) meg a DSSSL (Document Style Semantics and Specification Language), amelyek nyomtatáskor és egyéb output esetén határozzák meg, hogy miként is kell az adott dokumentumnak kinéznie. Egyelőre sem az egyik, sem a másik sem élvez elég támogatást ahhoz, hogy szélesebb körben is elterjedjen, és ha minden igaz, akkor most már nem is fog, mert mindinkább ismertté válik a cikk elején már emlegetett XML. Lehet, hogy az XML az ideális középutat jelenti a túlságosan korlátozott lehetőségekkel rendelkező, merev HTML, és a Web számára túlságosan bonyolult SGML között.

Az SGML leírása önmagában is majdnem ötszáz oldal, és ehhez jön további száz oldal függelék. Ehhez képest az Extensible Markup Language leírására elég egy alig 26 oldalas dokumentum. Valószínűleg már ezek az arányok is érzékeltetik, hogy az XML (különösen az SGML-hez viszonyítva) még akkor sem olyan rémisztően bonyolult, ha a HTML-hez hasonlóan szintén az SGML-nek (vagyis a jelölőnyelvek metanyelvének) egyik „leszármazottjáról" van szó.

Az egyik meghatározás szerint az XML lényegében arra szolgál, hogy „könnyűvé és egyértelművé tegye az SGML használatát a Weben", amibe persze beleértendő a különböző dokumentumtípusok definiálása éppúgy, mint az SGML-alapú dokumentumok megírása és Weben keresztüli továbbítása is — vagyis tulajdonképpen minden, ami annyira hiányzik a HTML-ből. Nem véletlen, hogy ugyanaz a World Wide Web Consortium (lásd a róla szóló cikket a 8. oldalon) fejlesztette ki, amely a HTML fejlődése felett is gyámkodik, és majdnem pontosan egy évvel ezelőtt, 1998. február 10-én fogadta el az XML 1.0-s ajánlását.

Az XML-ben az alapvetően nyitott és a saját jelölés definiálását is lehetővé tevő SGML-es filozófián túl az egységesítést és a következetességet szolgáló számos megoldás található, például a kis- és nagybetűk megkülönböztetése. A régi HTML ennek ellenére nem fog az egyik pillanatról a másikra eltűnni, és a megfelelő, ma még leginkább nem létező szoftverek segítségével majd konvertálni lehet a dokumentumokat XML-be. Persze, ha olyan tömegfogyasztásra szánt WYSIWYG HTML-editorral dolgoztunk, amely pongyola és következetlen HTML konyhanyelvet használt, akkor valószínűleg kézzel kell majd végrehajtanunk a változtatásokat. Cserébe viszont tényleg páratlan lehetőségek nyílnak meg előttünk. A megfelelően megformázott (valid) XML egyben SGML is, ezért a Weben kívül is széles körben használható és „újrahasználható", adatátvitelre alkalmasabb, könnyebben hozzáférhető és számítógéppel jobban feldolgozható dokumentumokhoz jutunk.

Ellentétben a HTML-lel, az XML terjedésének „motorjai" már nem annyira az egyéni felhasználók, nem az egyszerű webnauták lesznek. Erre az új eszközre sokkal inkább szükségük van a webes jelenlétben érdekelt nagy cégeknek. A natív XML applikációt, a MathML-t használó W3C-s Amaya browsertől kezdve a Microsoft Internet Explorer 4-es változatáig már ma egyre több webböngésző képes az Extensible Markup Language megértésére. Könnyen lehet, hogy az XML az Interneten ugyanolyan forradalmi változásokat indít el, mint annak idején a HTML, tehát a Web megjelenése okozott.

Galántai Zoltán



Hogyan bukkanhatunk új ügyfelekre meglepő helyeken? (Például az Internet segítségével!)

Azt már nyilván Ön is tudja, hogy az Interneten híreket közölhet cégéről, bemutathatja termékeit, felkeltheti potenciális vásárlói érdeklődését.

De mi lenne, ha honlapját kifejezetten eladásra használná?

Az IBM cégek ezreit segíti abban, hogy Internet oldalaikat forgalmas üzletté varázsolják, ahol az ügyfelek biztonságosan vásárolhatnak.

Röviden: megvalósítja az elektronikus kereskedelmet.

Bármilyen üzleti tevékenységet folytat, az IBM szakértő tanácsokkal, műszaki know-how-val nyújt segítséget Önnek, hogy honlapját dinamikus kereskedelmi csatornává formálja.

Évtizedes tapasztalatával az IBM készen áll arra, hogy a legmodernebb internet technológia alkalmazásával kösse össze cége honlapját és létfontosságú üzleti folyamatait. Integrálja a leltározás, számlázás, szállítás, raktározás műveleteit, hogy cége az e-business, az elektronikus üzleti tevékenység aktív résztvevőjévé váljon.

Látogassa meg www.ibm.com/e-business weboldalunkat, ahol megtalálja az üzleti tevékenység biztonságára vonatkozó e-business esettanulmányokat, tényeket, megoldásokat angol nyelven. Olvassa el magyar nyelvű tájékoztatónkat a www.ibm.hu címen, vagy hívja a 06 80 200 083-as zöld számot.

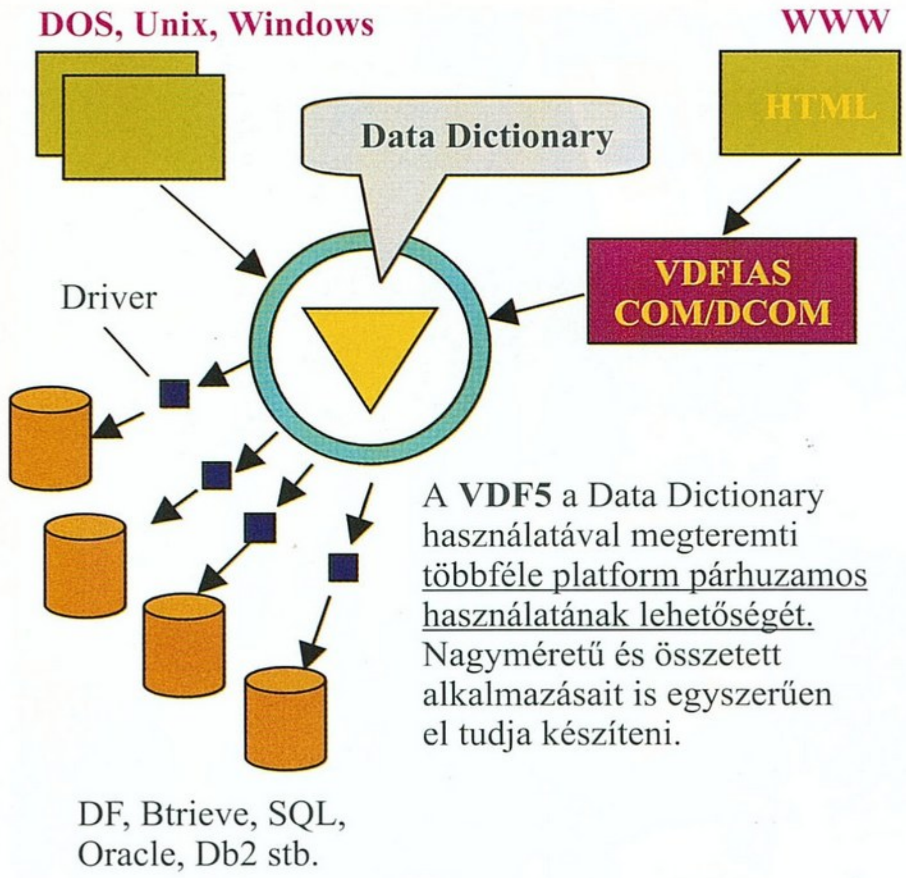
Magyarországi ügyfelek is igénybe vehetik a biztonságos elektronikus kereskedelem adta előnyöket az Inter-Európa Bank szolgáltatása révén. www.passage.ieb.hu

Hogy valóra váljék a biztonságos elektronikus kereskedelem (e-commerce), az IBM létrehozta a világ első SET 1.0-án alapuló kereskedelmi szerverét, a Net.Commerce-t.

e-business

IBM

Nagy megoldások egy kis bolygónak



A **VDF5** a Data Dictionary használatával megteremti <u>többféle platform párhuzamos használatának lehetőségét.</u> Nagyméretű és összetett alkalmazásait is egyszerűen el tudja készíteni.

A Visual DataFlex alkalmazhatósága

Forgalmazza: **NEXT Software Kft**
1119 Budapest XI., Andor u. 60.
Telefon: 208-4643, 208-4631
E-mail: nextsw@hungary.net
Weblap: www.dataobject.hu/dfklub/





A KIMSOFT februári ajánlata

Akció (amíg a készlet tart)

Borland Delphi 3.0 Standard	23 200,-
CorelDRAW 8 CD /Upgr.	**77 ~~0~~00,-/69 900,-**
CorelDRAW 5.0 CD	28 900,-
Windows NT 4.0 Server+10 CAL	**154 900,-**
Norton Antivirus 5.0	11 996,-/5 996,-

Szoftver újdonságainkból

Adobe Illustrator 8.0 /Up.	**131 400,-/47 900,-**
Borland C++ Builder 3.0 Prof. Up.	82 400,-
Borland Delphi 4.0 Standard	**33 900,-**
Cég-Kód-Tár (350 000 magyar cég)	10 000,-
Corel Print Office magyar	**21 400,-**
IntelliCAD 98 for Win32	110 000,-
Kai's Super Goo	**18 400,-**
Kulcs-Soft Kereskedelmi rsz. Win.	60 000,-
MathCAD 8.0 Prof. (International)	**109 900,-**
MS Plus! for Win98	9 400,-
PaintShop Pro 5.0 for Win95	**23 400,-**
Partition Magic 4.0	20 500,-
„Sikító titkárnő" Win32 (alap + k. h.)	**6 800,-**
Visual Basic 6.0 Prof./Up.	119 900,-/60 900,-
Visual C++ 6.0 Prof. /Up.	**119 900,-/60 900,-**
Visual Studio 6.0 Prof./Up.	244 800,-/121 996,-
Windows 98 /Upgrade	**46 400,-/24 400,-**

CD-ROM-ok, játékprogramok

ChessMaster 6000	**8 900,-**
Dark Reign /European Air War	7 400,-/8 492,-
FIFA 99 /Mech Commander	**9 800,-/9 400,-**
Angol-magyar nagyszótár CD-n	**15 400,-**
Learn to Speak English - 2 CD (Akció!)	15 996,-
Lopva Angolul 1. /2. /3. (egyenként)	**4 750,-**
Manó Élővilág /Manó Matek	4 750,-/4 750,-
Matematika /Fizika felkészítő	**4 800,-/4 800,-**

Adobe PageMaker 6.5	184 900,-/54 900,-
Adobe PhotoShop 5.0 magyar	**209 900,-**
ARJ 2.6 /PkZip 2.04 DOS	13 400,-/14 000,-
AutoCAD LT 98 /Up.	**109 900,-/26 400,-**
CA-Clipper 5.3+Tools 3.0+VO 2.0	57 900,-
Check It 98 Diagnostic Suite	**39 900,-**
CleanSweep 4.02	12 400,-
Close Up 6.5 Dual Pack	**38 900,-**
Corel Gallery 205 000	15 200,-
Eudora Pro 4.0 for Win95	**16 400,-**
F-Prot 4.02 Prof. (antivirus pr.)	Hívjon!
Harvard Graphics 98 Comp. Up.	**30 600,-**
IBM VisualAge for JAVA Prof.	26 400,-
Lotus Freelance 97 for Win95	**18 200,-**
Lotus Organizer 97 for Win95	18 200,-
Lotus SmartSuite 97 C. Upgrade	**55 400,-**
Magyar Fontok '97+ (4000 font)	5 200,-
Mapple V Release 4 (Akció!)	**41 200,-**
McAfee Office (6 szoftver együtt)	23 800,-
McAfee VirusScan 4.0 Deluxe	**14 900,-**
MS FrontPage 98 /Up.	34 900,-/12 600,-
MS Office 97 /Upgr.	**110 900,-/56 400,-**
MS Office 97 Prof /Up.	132 600,-/69 600,-
Modem Share32	**44 000,-**
Norton Antivirus 5.0 magyar	16 900,-
Norton Commander 2.0 Win32	**9 996,-**
Norton Utilities 3.0 Win95 magyar	18 400,-
QEM 97 v9.0	**20 500,-**
QuarkXPress 4.0 for Win95	**221 400,-**
SuperPrint 5.0 for Win95	17 700,-
System Commander 4.0 Deluxe	**23 400,-**
Visio Prof. /Techn. 5.0	108 600,-/108 600,-
Windows Commander 3.5	**11 600,-**
WinFax Pro 9.0 Win95	29 000,-/14 600,-

A közölt árak nem tartalmazzák a 25%-os áfát, és a helyszíni üzembehelyezés költségeit.

Teljes árjegyzékünket kérje faxon tone üzemmódban a faxbankból: 2-333-666/1497#

KIM-SOFT Számítástechnikai és Kereskedemi Kft.
1112 Budapest, Hegyalja út 70. fszt. 2.
Telefon: 319-8973, 319-8967 Fax: 319-9760

Kritikai vitriológia

Amit nem szeretek...

Hangulatjelentés éjjel háromnegyed egykor, avagy: mit NEM szeretek a PC-s számítástechnikában? Először is, nem szeretek ilyenkor, ihlet nélkül cikket írni. Azonban gondolnom kell hűséges olvasóimra, még hűségesebb főszerkesztőmre... nem hagyhatom őket cserben. Persze a rovatomnak szánt két oldal abban az esetben sem jelenne meg üresen, ha ez a cikk most nem készülne el idejében (az új hirdetők sorban állnak, hogy végre bekerülhessenek a lapba), de ez a bevezető talán magyarázatot ad önkényes és ösztönös témaválasztásomra.

Savazás lesz! Nem kell megijedni, nem népsavazás, azt meghagyom magyar származású belga segédlelkészeknek és családtagjaiknak.

Nem szeretem általában

... a számítástechnika misztifikálását, amikor a hozzá nem értők varázslónak nézik a témában mindennapos otthonossággal mozgókat.

..., hogy a Microsoft Varázslónak (Wizard) nevezi a hozzá nem értők számára készített szoftveres uniformisokat.

... az uniformisokat sem.

..., hogy a PC-s profik lenézik a gyakorlatlanokat, noha csak nagyobb gyakorlatuk van az egér tologatásában, és tudják, mit jelent PC-s szakzsargonban, hogy default. Az error szó láttán viszont ijedtség helyett szakszerűen káromkodnak.

..., hogy a mindentudó jóslatok szerint az emberek életszínvonalát emeli, életminőségét jobbítja majd a számítástechnika hétköznapivá válása.

... a számítástechnikai vásárokat, immár sem belföldön, sem pedig külföldön. Bár utazni élvezet, de megalázó sok kilónyi prospektussal, repi CD-vel, floppyval, tyutyumutyu üveggyöngygyel felmálházva vonulni pavilonról pavilonra. Találja meg az utat hozzám magától (vagy az ügyes marketingesek révén) a számítástechnika mindazon vívmánya, amelyre szükségem lehet valaha is az életben.

... a számítástechnikai szaksajtót (magamat is beleértve), mert a legritkábban célja — tisztelet a kivételnek —, hogy olvasói érdekében teszteljen, tárjon fel hibákat, hiányosságokat, mutasson rá határozottan a svindlerekre és svindlijeikre. Ehelyett a médiákban a hirdetők alázatos kiszolgálása és azok ülepének nyelvvel történő illetése folyik — ismét tisztelet a kivételnek —, mialatt a futószalagon gyártott, fellengzős, agyontömjénezett sajtóanyagokból — megint tisztelet a kivételnek —, felületes, semmitmondó fércművek készülnek az olvasók sérelmére.

... azokat, akik nem veszik a lapot, és azokat sem, akik veszik az olyan lapot, amely külföldi sablonok alapján készül. Az átvett külföldi — ott talán bevált — cikkeknek se füle, se farka, hasznosíthatósága pedig arányos a lapméret csomagolásra való alkalmasságával.

... az öntudatos szoftverkalózokat.

... a nemzetközi és a magyar szerzői jog kiforratlan alkalmazását a számítógépes szoftverek világában.

... a szoftverfejlesztők és szoftvergyártók nullával egyenlő garanciavállalását.

Nem szeretem hardverice

... a 103-nál több gombos billentyűzeteket, mert nem akarok négykezest játszani a klaviatúrán.

... a miniatűr noteszgépek egérpótló pozicionálóit, a mikromanipulátorokat.

... a kábelrengeteget az asztalon, asztal alatt és mellett, mert nem akarok Laokoón-csoportvezető lenni.

... a tápegység és a processzor hűtőventillátorának folyamatos zaját. A PC alapzaját. Találja már ki végre valaki a hangtompítós PC-t!

... a lomha hagyományos floppyegységeket. Lassú víz part(ició6óó)ot mos.

... a hasonlóképpen lassú, körülményes adatarchiváló egységeket (DAT, JAZ, ZIP, CD-író stb...)

..., hogy a DVD még mindig nem jutott el, nemhogy a csúcsra, de odáig sem, hogy érdemes legyen róla ábrándozni.

F-Secure

Biztonság minden téren

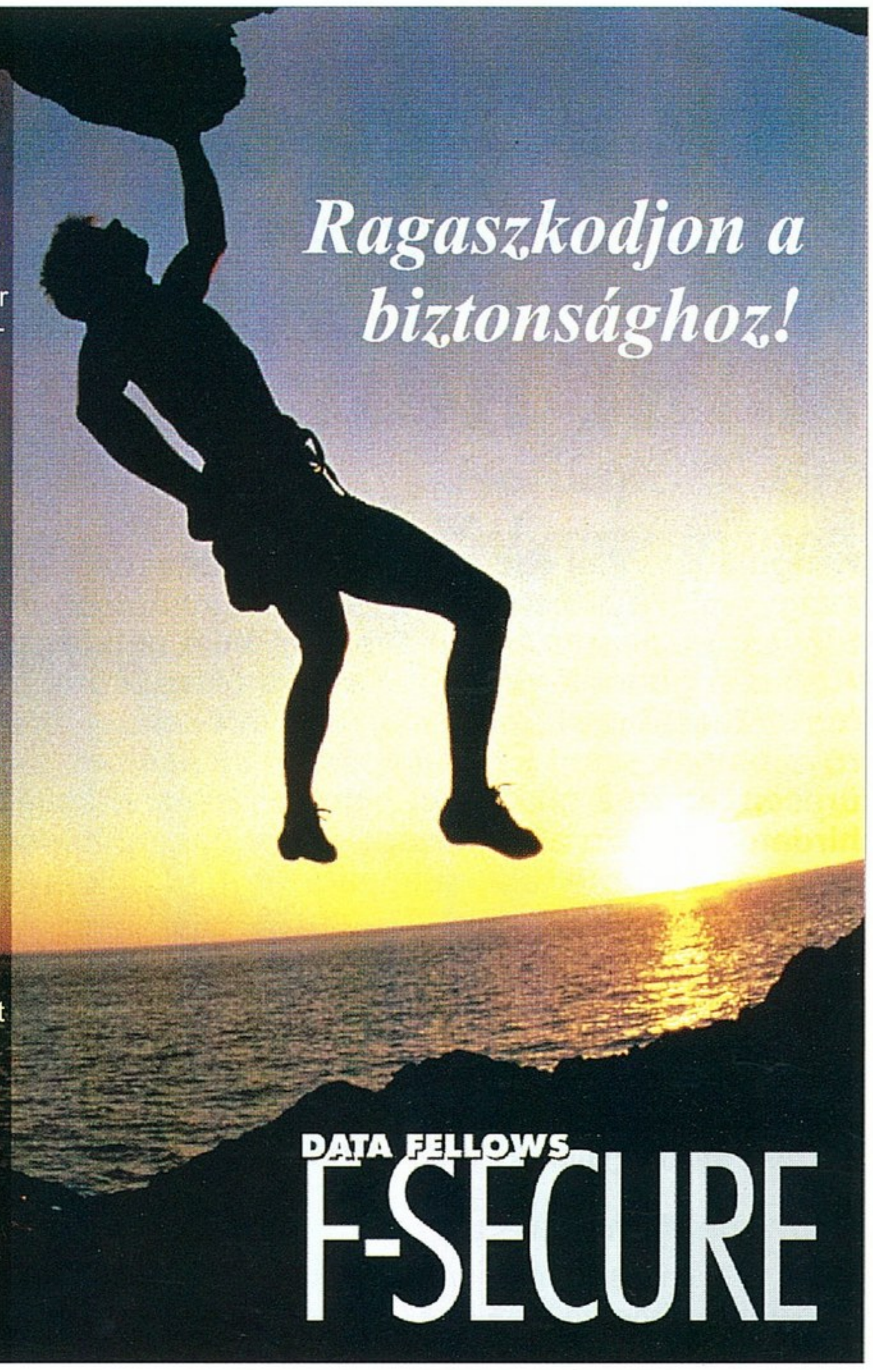
Az F-Secure rendszer az egyetlen adatbiztonsági csomag, amely képes a vállalaton belül felmerülő minden biztonsági feladatot megoldani. A rendszer elemei a hordozható számítógépektől kezdve az asztali gépeken át a legnagyobb szerverekig minden gépre telepíthetők, majd a központi, házirend-alapú menedzsment eszközzel vezérelhetők. Segítségével egyetlen rendszerbe integrálva történik a vírusellenőrzés, a hálózati forgalom titkosítása és a gépeken tárolt adatok titkosítása.

Az **F-Secure Anti-Virus** a CounterSign™ technológiára épülő, moduláris víruskereső rendszer, mellyel egy időben több víruskereső mag használható. A Gatekeeper™ technológia segítségével minden használatba vett, avagy az Internetről letöltött állomány azonnali ellenőrzésre kerül.

Az **F-Secure FileCrypto** valódi valósidejű, fájlszintű titkosításra képes. Európai fejlesztésű program, maximális biztonságú, szilárd titkosítást alkalmaz, 256 bites kulcshosszal. Teljes hálózati adminisztrációs lehetőséggel rendelkezik: automatizált telepítés, központi kulcsvisszanyerés (key recovery) és többfelhasználós működés is lehetséges.

Az **F-Secure VPN+** Internet-szabványokon alapuló, a hálózati forgalom titkosítására szolgáló eszköz. Alapja az IPSec protokoll és az IKE (Internet Key Exchange), kompatíbilis minden szabványos kulcsközponttal és kulcshitelesítő infrastruktúrával. Képes kliens-kliens, kliens-szerver és szerver-szerver közötti kapcsolatok kialakítására, így minden elképzelhető VPN (Virtual Private Network) topológia kiépíthető vele.

Cégünk az F-Secure rendszer minden eleméhez teljeskörű támogatást nyújt felkészült csapata révén, mely a gyártó cég **Certified Anti-Virus Center**



2F 2000 Számítástechnikai és Szolgáltató Kft.
http://www.2f.hu/ e-mail: info@2f.hu
Tel: 212-7141, 212-7142 Fax: 212-7143

INFORMÁCIÓKÉRÉS: 01 ▲

KeSzo Kft

1055 Budapest V., Falk Miksa u. 6.
Telefon: 332-8717 *Fax: 302-5136*
E-mail: sales@keszo.com *Web: www.keszo.com*

PkZip 2.5 Command line, UUencode...	12.000	MS Office 97 magyar prof. / upg.	134.000 / 70.000
Britanncia Encyclopedia / MS Encarta 99	33.000 / 14.000	WinFAX Pro 9.0 NT, Win95 / upg.	36.000 / 15.000
Windows Commander 3.52 16/32bit (magyarul is)	11.000	Hálózatos faxkezeléssel!	
FAR 1.60 / RAR 2.05 / ARJ 2.61	10.000 / 10.000 / 18.000	Partition Magic 4 (particionálás adatvesztés nélkül)	23.000
WinZip 7.0 / WinARJ	14.000 / 18.000	Visio 5.0 Win95/NT Standard / upg.	56.000 / 36.000
F-Prot Professional	43.000	Visio 5.0 Professional Win95/NT / upg.	112.000 / 52.000
Norton Antivirus 5.0 magyarul is	20.000	Visio 5.0 Technical Win95/NT / upg.	112.000 / 52.000
Clarion Prof. 5.0 / upgrade	174.000 / 58.000	Photoshop 5.0 Win95/NT / upg.	232.000 / 68.000
Hot Metal Pro 5.0	36.000	Photoshop 4.0 Win95/NT magyar! / upg.	184.000 / 69.000
MS FrontPage 98	33.600	NT 4.0 Server / WKS Resource Kit	35.400 / 16.200
MS Project 98 / upgrade	112.000 / 44.900	Win 98 Resource Kit / Office 97 Res. Kit	16.000 / 14.000
Norton Utilities 3.0	20.000	Norton Commander 1.2 Win95/NT / upg.	20.000 / 11.000
System Commander 4.x Deluxe	30.000	Adobe Acrobat / Corel ArtShow 7	82.000 / 9.900
Norton Uninstaller	16.000	Multikey 3.5 / upgrade	4.000 / 2.000
MathCAD 8.0 Plus	182.000	NT Key 4.0 / upgrade előző verziókról	10.000 / 6.000
Procomm 4.7 Win95/NT Internet, fax, modem	51.000	Adobe Illustrator 8.0 / upg.	170.000 / 59.000
DriveImage (FAT16/32, HPFS, NTFS)	24.000	QuarkXPress 4.0 PC/MAC / 3.32 PC	256.000 / 225.000
DiskClone from Quarterdeck HDD copy!		Helyes-e? for QuarkXpress 4.0	59.000
Adobe Type Manager 4.0 deluxe for NT	26.000		
MS Office 97 magyar / upg.	112.000 / 47.000		

Áraink áfa nélkül értendők!

INFORMÁCIÓKÉRÉS: 13 ▲

..., hogy egy ma vásárolt csúcsszuper hardver értékének felezési ideje lassan egyenlő lesz az egykilós kenyérvekni felezési idejével, ha kellően éles a késünk.

... a PC-s körökben árult kommersz mikrofonok csapnivaló felvételi minőségét.

... a megalomániás hangkártyákat. Dolby-sztereo-surround a már említett alapzaj mellé. Esetleg használjunk hozzá fej- vagy fülhallgatót? Stúdió minőségben.

... a töméntelen CD-t, ami folyamatosan felhalmozódik egy piacorientált PC-felhasználó háztartásában. Egyáltalán reciklálható?

Nem szeretem szoftverice

... hogy egyre inkább minden Microsoft. Semmi kifogásom nem volna ellene, ha ez a márka egyúttal a minőség védjegye is lenne.

... hogy a felhasználók 95%-a olyan programokat használ, amelyek tudásának jó ha 5%-át ismeri és hasznosítja. Aki csak leveleket akar írni, annak minek vennie (vagy minek adjanak alá) egy Kossuth Nyomdát?

... a DOS egyre gyengülő támogatottságát az újabb oprendszerek szintjén, s hogy az új, csupán pár éve a PC-s világba bekerülők már meg sem ízlelhetik a DOS-os programok fürgeségét, lényegre törő eleganciáját.

... az egyedi adatformátumokat, az egyes alkalmazások és platformok közötti információ- és adatcsere ellehetetlenítőit.

... az ékezetes karaktereket nem (normálisan) kezelő alkalmazásokat. Nem csupán magyar probléma, bűzlik ez Dániában is.

... a csupán menetrendszerű, de valódi plusz értéket nem nyújtó programverzió-váltásokat. Be kellene tiltani a léptetőmotorokat!

... az egyénieskedő billentyűkombinációkkal operáló szoftvermegoldásokat. Ha már Windows, legyen a megszokott. De nem lúd, így ne legyen kövér.

... az uninstallálhatatlan, kiirthatatlan programokat. Felveri a gaz a gépeinket.

... az újratelepítéseket, a ki- és visszatelepítéseket, az ezekből fakadó adatvesztéseket.

... az 1,44 MB + 16 bájt hosszúságú, egy floppyra éppen rá nem férő telepítőprogramokat.

... a tetűlassan ébredő operációs rendszereket. Nem lehetne ott folytatni, ahol a legutóbbi kikapcsoláskor befejeztem?

..., hogy a harddiszken az állományok széttördelése az oprendszer szintjén keletkezik, de külön program kell hozzá, hogy ezt a „lassítósztrájkot" később megszüntessük. Tipikusan az „elgáncsolom, utána felsegítem" esete.

... a lassuló, monstrum méretű programóriásokat. Táguló világegyetem? Igen, de a sötétség is fénysebességgel terjed!

... a többcédés, CD-csereberélős programokat. Lemezlovasképzés?

... a giccsparádévá avanzsáló képernyővédők sorát. Alkalmi jópofiságnak még elmegy, de minek kellene alapjáratban hagyni autónkat parkolás közben.

... az adatarchiváló programok oktondi módon kényelmetlen kezelhetőségét.

... a semmitmondó hibaüzeneteket és az állandó fagyásveszélyt. Ki látott már PC-t fagyállóval feltöltve?

... a valóságtól elrugaszkodott szoftverárakat. Lassan egy MS Office Professional ára megegyezik azon gép árával, amelyiken futtatjuk. És egy számítógépes szótár mitől drágább, mint a nyomtatásban is megjelent változata?

Nem szeretem internetice

... a sávkeskenységet, ami ugye a sávszélesség ellentéte.

... az „internetszolgáltatókat".

... az internetezőket általában, mert a Netet terhelve engem is lelassítanak.

... a bő lére eresztett e-maileket. A múltkor is kétszer kaptam volna meg Tolsztoj Háború és békéjét, de az óriásbébi már a mailserveren elvérzett, a headerből értesültem a túlméretről (és a küldő — nevezzük továbbra is csak Tolsztojnak — kilétéről), így azonnal, letöltés nélkül búcsút mondtam neki.

... a két attachmentnél több mellékletet tartalmazó elektronikus leveleket. Csomagkapcsolt hálózat?

... a színes, animált, HTML-lel spékelt e-maileket. Megannyi levélbomba.

... azokat az e-mail kliens szoftvereket, melyeket nem lehet gyorsan és egyszerűen, bármilyen kritériumszerintviszszakeresni.

... a mailserveren lévő adatokat menedzselni képtelen e-mail klienst.

... a fentiek miatt egyelőre a teljes Microsoft Outlook családfát. Komoly generációs problémákat vélek felfedezni köztük.

... az állandóan ismétlődő képanyaggal működő szex-site-okat. Mintha a Playboy éveken keresztül mindig ugyanazzal a képpel jelenne meg a címlapon. Igaz, így az öregedés is kizárva — papíron.

... a sok szemét, karbantartatlan információt. Az elaggott kültéri hirdetőtábla esete, amelyet már csak az időjárás szerkeszt.

... a háttérben adatbáziskezelést folytató, nehézkes webhelyeket.

... a fantáziátlan, túlcicomázott weboldalakat, ahol elvész, elsikkad maga az információ.

... az ezerlépcsős, agyonszegmentált felépítésű homepage-eket.

... a képi humor szinte teljes hiányát.

... a rengeteg helyesírási és stiláris hibát, ízlésficamos megnyilvánulási formákat, amik ellen az olvasó számára nincs védelem. Ha már olvasom, és benne vagyok a kelepcében, usgyi, gyorsan el onnét.

... hitelkártyám állandó felemlegetését, meg hogy akinek ilyenje nincs, az nem is teljesértékű netizen.

... a nagy lendülettel induló weblapokat, amelyek gazdája később nem akarja még magának sem bevallani, hogy ő is, műve is kifújt. Mementóként ott éktelenkedik, hogy az utolsó módosítás dátuma Kr. e. 600... Ó testamentum.

... a 2000. évváltással kapcsolatos, többnyire átlátszóan felelőtlen megfelelési ígéreteket. Nem baj, hamarosan kinőjük.

..., hogy az információ nem pontos, nem fontos.

Mégis ebből élek én is. Lehet, hogy nem vagyok egyedül?

Herczeg József

Clipper és a dátumkalamajka

Kedves Alaplap!

Már kiskorod óta vásárollak, amikor még Mikromagazinnak becéztünk ;-) Ha jól összeszedem magamat és példányaidat, talán minden számodat fel tudom mutatni (bár nem vagyok előfizető, mert kiemelnének a postaládámból). Megjelenésedkor átfutlak, és amint jut rá időm, végig ki is olvaslak. Talán ebből is látszik, hogy adok a véleményedre, és bízom benned.

A Kritikai vitriológiát különösen kedvelem, és pont ott ért egy bizalmamat csorbító meglepetés. Az 1998. októberi számban olyan megállapításokat találtam, hogy az szerintem a valóság elferdítése. A Clippert lehet szeretni, és lehet utálni, ahogyan az más, akár a legmenőbb (leggazdagabb) cég termékeivel kapcsolatban is gyakran tapasztalható. Néhány dolgot azonban nem szabad megtenni:

— Nem szabad egy programnyelvet kizárólag kezelőfelületének korszerűsége, sőt divatossága alapján megítélni.

— Nem szabad egy programnyelvet a benne (vele) dolgozók produktumai alapján minősíteni.

— Nem szabad téves ismeretek alapján ítélkezni.

Sorrendben:

A karakteres (DOS-os) felület ma már nem divatos, de hatékony és gyors programfutást eredményez, még olyan gépeken is, amilyeneken az éppen aktuális grafikus oprendszer el sem indítható.

A Clipper gyors és hatékony programok fejlesztését tette lehetővé a maga idejében. Nem szabad elfelejteni, hogy a dBase II már 1981-ben kereskedelmi forgalomban volt, és az 1988-as dBase IV változatot még most is elterjedten használják. Persze a mai „tedd ide – tedd oda", grafikus, kódgenerátoros feljesztőkörnyezetek korában már sok minden másként van. De azért a Basic-ben is rengeteg hozzá nem értő írogat programot, mégsem szólja le senki a nyelvet a produktum alapján. Clipperben is vannak jó és szép, grafikusan is látványos programok (például térinformatikai rendszer... ja és ez persze magyar fejlesztés), valamint csapnivaló termékek is.

Az eddigiek többé-kevésbé a ízlések és pofonok témakörébe tartoztak, de a tény az tény. Tessék egy dbf-et valamilyen editorral megnézni. A dátum típusúnak definiált mezők bizony ééééhhnn formában vannak tárolva. Ez persze nem a Clipper kigondolóinak, hanem a dBase fejlesztőinek érdeme. A fejlesztő cég (ha jól tudom) az Ashton-Tate volt, és a dBase III, azaz 1983 óta van érvényben ez a formátum. A Clipper-ben létezik továbbá két utasítás, melyekkel a probléma nem túl nagy erőfeszítés árán megoldható. A SET EPOCH éééé utasítással a viszonyítási időpontot a megadott évhez lehet rögzíteni, de ez persze csak az évek kétjegyű kezelésénél érdekes. A négyjegyű megjelenítést a SET CENTURY utasítással lehet bekapcsolni. A részletek a megfelelő webcímeken megtalálhatók (van magyar is). A Y2K probléma tehát a forráskód ismeretében teljes mértékben megoldható. Az egészet nem is a Clipper védelmében tettem szóvá, hanem a szavahihetőség miatt!

Harsán Péter

A Kritikai vitriológia műfaji sajátosságából adódik, hogy annak megállapításai gyakran tartalmaznak túlzásokat, poénkodásokat, csipkelődéseket. A humor igazsága aligha lenne bármikor is megvédhető, ha „az üzenet" lényegének felfogása helyett a mondanivaló hangsúlyozására alkalmazott stiláris elemeket vagy szövegeket vetnénk alá komolykodó, tételes elemzésnek. Túlzás tehát úgy felfogni, hogy azok a mondatok Herczegnek (vagy pláne lapunknak) a Clipper programozási nyelvről alkotott szakmai ítéletét tartalmazzák.

Tény viszont, hogy a mai felhasználók egeret akarnak tologatni grafikus képernyő előtt, és gépelés helyett vágólapról szeretnék behívni a szövegrészeket. Ha a kódolandó feladat ugyanaz marad is, a külalak legyen más. A fő csapásirány pedig a DOS-os Clipperből irányul a 32 bites windowsos alkalmazásokba, lehetőleg az újbóli rendszertervezés elkerülésével. Ahogy azt olvasónk is pontosan látja, ebben az esetben a forráskód megléte elengedhetetlen. Az is tény viszont, hogy a 2000. évi dátumkezelés kapcsán Herczeg igazságtalanul „csapott oda" a Clippernek, de ezt a hibát az 1999. januári hónap témájában Báró Csaba cikke remélhetőleg ellensúlyozza.

A DÁTUMKALAMAJKA összeállításra is kaptunk már kiegészítő észrevételeket, melyek csak megerősítik mondanivalónk lényegét, hogy a számítástechnikai szakma sajnos nem a tőle elvárható gondossággal és nem kellő időben cselekedett annak érdekében, hogy a számítástechnikai eszközök (hardver és szoftver) a 2000. év utáni dátumkezelés során ne okozzanak gondot (és extra költségeket!) a felhasználóknak. És itt most nagyon finoman fogalmaztunk, ahhoz képest, amit a 600, 900 vagy ki tudja hány milliárd dolláros sarc érdemelne...

Egy másik, elkésett észrevételt éppen a fenti Kritikai vitriológiában megemlített sokféle dátumformátum egyike váltotta ki. Szondi Egon János kollégánk a következőket írta „széljegyzetként": „Az YYDDD forma némi nosztalgiát ébresztett bennem. A hetvenes évek elején egy (nem klón) IBM géppel volt dolgom. Az IBM System/360 DOS ugyanis így jelölte a dátumot: 1973. január 1 = 73001, december 31 = 73365. Ennek már 25 éve..." Hát igen, ez így 5 bájt, amihez hozzátéve az évszázad két bájtját 7 bájton és decimálisan is elfért volna a teljes évszám és az év megfelelő napja, dátumszámításokra alkalmas formában, időszámításunk kezdetétől 9999-ig jól használhatóan... sőt maradt volna egy tartalék bájt is, a pesszimistáknak az időszámításunk előtti dátumok kezelésére, az optimistáknak 10 000. január elsejétől egészen 99 999. december 31-ig...

(A szerk.)

CD-kritika

Az 1999. januári számban a CD átalakulását beharangozó cikk alapján (21. oldal) kíváncsian és örvendezve kezdtem hozzá az új termék tanulmányozásához. Ahogy megyek végig a szokásos rovatokon egyszercsak azon kezdek gondolkozni, hogy miért nézem 15 cm-ről a képernyőt. Hamar rájöttem a válaszra: AZÉRT, HOGY LÁSSAM A BETŰKET!

Ugye ezt ti sem gondoltátok komolyan?

— Fekete alapon a karamell betűk elég jól néznek ki.

— Világoskék alapon a fekete betűk még elmennek.

— Sötétkék alapon a világoskék betűk... ám legyen.

DE: Szürkéskék alapon fekete betűk... Plíííz! Nem akarok megvakulni. (Ráadásul ebből volt a legtöbb sor.)

Rögtön a PC-X néhány korábbi pepita cikke jutott eszembe, szürke-lila pöttyös háttérre ültetett UV-zöld-piros betűkkel. Na jó, de az a lap a fiatalabbakhoz szól, akiknek még van néhány évük a teljes vagy részleges megvakulásig.

Nagyon kérlek benneteket, hogy ilyet ne csináljatok! Aki azt akarja, hogy mások is elolvassák írását, az vagy sötét alapon világos, vagy világos alapon sötét betűket használ. Remélem, tudjátok, mi a teendő.

Hittl Szilárd

Igen, tudjuk, el is végeztük az első korrekciókat, és a későbbiekben sem habozunk változtatni azon, ami nem jól sikerül. Az első bírálók zömmel formai észrevételeket tettek. Hittl Szilárd fogalmazott a legkeményebben, de teljesen megalapozottan, ezért is idéztünk az ő leveléből.

(A szerk.)

A hónap témája

A weblapok szerkesztéséhez, illetve az Internet használatához alkalmazható programok gazdag választékából igyekeztünk CD-mellékletünkön olyan összeállítást közreadni, hogy abban mindenki találjon számára hasznos anyagokat.

A CD-n található weblapkészítő programok egy részét a hónap témája összeállítás cikkei is megemlítik. A webszerkesztők között vannak olyanok, amelyk főleg egyszerűbb és rövidebb dokumentumok készítésére alkalmasak, mint például a DiDaPro (Faico Information Solutions). Vannak továbbá segédprogramok HTML oldalak formájának vagy színvilágának módosításához (Font and Body Tag Wizard — Jeff Carpenter). A szabadon hozzáférhető programok között találunk azonban olyanokat is, amelyek használatával professzionális szinten lehet a weblapot elkészíteni. Ilyen például az Allaire Corporation HomeSite programja, amelynek sokoldalúságára jellemző, hogy Word dokumentumból át tud venni táblázatokat a vágólapon keresztül.

A weblapok grafikai kialakításához nem nélkülözhetőek a megfelelő tudású programok, de néha elegendő egy-egy nyomógomb vagy háttérpanel gyors beillesztése is. Ilyen esetekben olyan egyszerűbb segédeszközök is jó szolgálatot tehetnek, mint például a Buttonz & Tilez program (Björn Ischo készítette).

A CD-n lévő webes programok, segédeszközök egyik jelentős csomagja az Adobe cég kollekciója, benne a PDF formátum megjelenítésére alkalmas segédprogramokkal és a programozóknak szánt kiegészítésekkel. Az előbbi csoport az Acrobat Reader különböző platformokra készült verzióit öleli fel. Ezekből a CD-melléklet Windows és Linux platformon készült dokumentumainak olvasására alkalmas verziók már korábban is rajta voltak a CD-n, most ez kiegészült a DOS-tól a különböző Unix verziókig több más operációs rendszer PDF változataival. A Weben fellelhető PDF dokumentumokat olvasóink ezáltal szinte valamennyi platformon elérhetik.

Az Adobe az általános publikációs programokon kívül a webes szerkesztéshez is készít programokat, így az Adobe PageMill 3.0 szabadon kipróbálható verziója szintén megtalálható a CD-mellékleten. Ez a próbaváltozat a telepítést követő 15 napon át teljes funkcionalitásában mutatja be a PageMill képességeit. Telepítése kb. 20 MB lemezterületet, 16 MB RAM-ot és 32 bites Windowst igényel.

A nem Macintosh fejlesztőkörnyezethen futó alkalmazásokhoz is vannak Adobe programfejlesztési segédletek (Software Development Kit — SDK). Ezek közül a PageMill mellett számos más Adobe SDK-t is felraktunk a CD-mellékletre.

Egy további SDK, de immár a Microsofttól, a dinamikus webdokumentumok készítését segítő készlet. A Microsoft Dynamic HTML Editing Component SDK lehetővé teszi az alkalmazásfejlesztőknek és a webdokumentumok készítőinek, hogy a WYSIWYG jellegű DHTML szerkesztés képességét beépítsék a website-ba vagy az alkalmazásba. Ennek alapja, hogy a szerkesztő komponens a Microsoft Component Object Model (COM) technológiáján alapul. Az Internet Explorer telepítőkészlete egyébként tartalmazza a FrontPage Express programot is. Az MS DHTML szerkesztési lehetőség a 32 bites Windows objektumrendszerét használni tudó más programnyelveken fejlesztők rendelkezésére is áll. (Például a Microsoft Visual Studio csomagjában található fejlesztőeszközökkel vagy az Inprise Delphi, illetve a C++ Builder eszközökkel dolgozóknak.)

Egy másik programozási segédeszköz-gyűjtemény a különböző scriptek kezelését segítő komponensekből áll. Ezek alapja a scriptek futtatókörnyezetét biztosító motor, a Microsoft Script Control (SCT10EN.EXE). Részben ezen a technológián alapul az Intenet Explorer böngészőprogram VBScript és JScript kezelése is. Ehhez tartozik egy frissített (a CD-n második publikus béta-változatában megtalálható IE 5-höz készült) futtatóeszköz, a SCR50EN.EXE, melynek teljes VBScript dokumentációja is telepíthető (VBSDOC.EXE). Az újabb Windows platformok kínálta lehetőségek kiaknázására a Microsoft elérhetővé tette a Windows Scripting Host rendszert a különböző nemzeti Windows verziókhoz is (WSHEN.EXE, WSHHU.EXE), azok használatát pedig külön mintagyűjtemény segíti (SAMPLES.EXE).

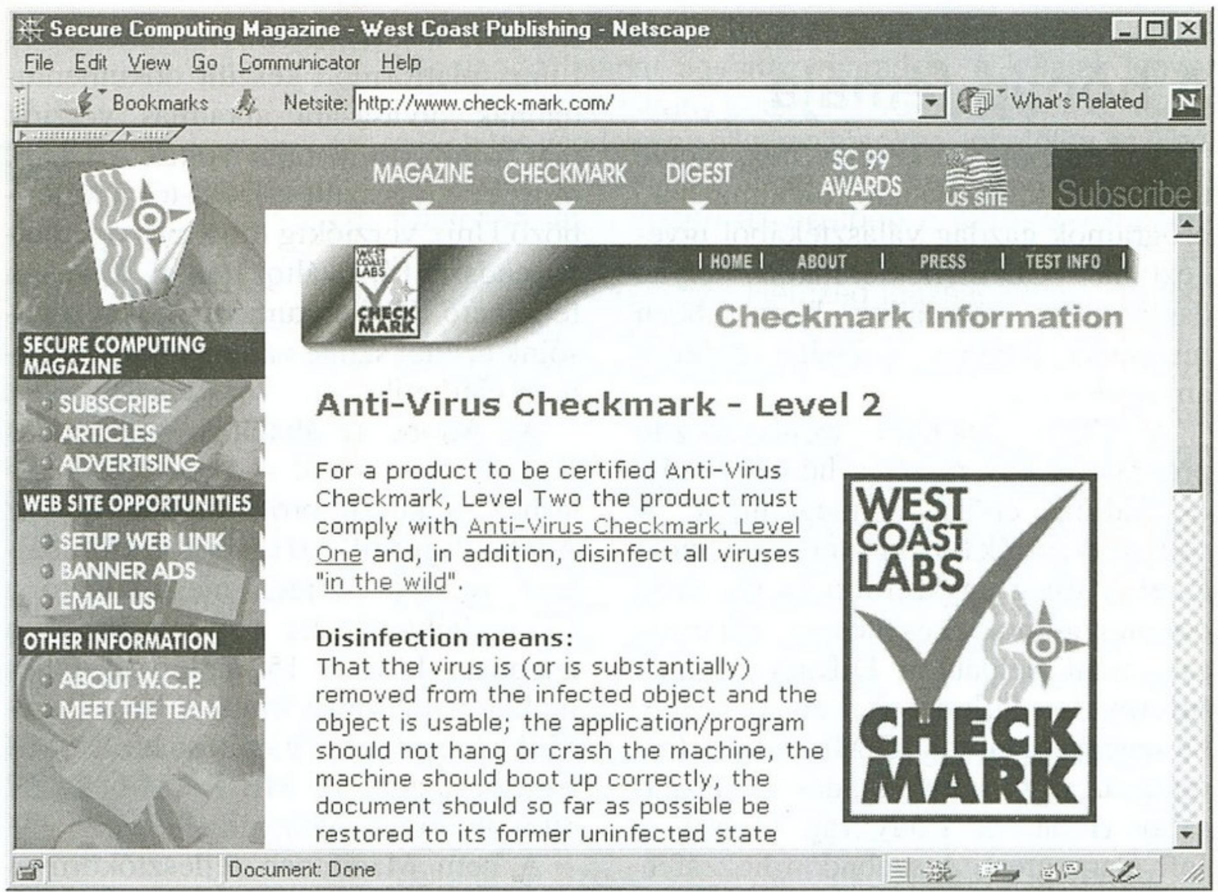
A nagy szoftvergyárak tevékenységében található réseket nem egyszer a kisebb fejlesztőműhelyek tömik be, vagy shareware koncepció jegyében készült vagy teljesen szabadon felhasználható, hatékony kis programokkal. Ilyen például a Moon Software Bookmark Wizardja, amellyel az Internet Explorer Favorites állományát átalakíthatjuk egyszerű HTML-dokumentummá, ezáltal az bármilyen más böngészőből elérhetővé válik.



Euro

A teljes egyesülés felé tartó Európa 1999. január 1-jével bevezetett új pénzneme, az euro közvetlenül érinti az informatikai feladatokat is, hiszen a számítástechnikai rendszereknek kezelniük kell részben a pénzügyi adatokat, részben magát az euro jelet, mint karaktert. A Win95 és NT operációs rendszerekben ehhez szükséges kiegészítéseket CD-mellékletünkre is felraktuk. A Windows 98 karakterkezelése szintaktikailag már ismeri ugyan a szükséges billentyűkombinációt (ALT+128), de True Type fontkészletei még nem tartalmazzák a karakter rajzolatát. Aki tehát Win98-at használ, annak hasznosak lehetnek a mellékletünkön megtalálható javított betűkészletek. Az Adobe-tól viszont frissített Type 1 fontokat tölthetnek le a CD-ről, ha PostScript rendszert kívánnak euro jellel kiegészíteni.

Simay Endre István



Microsoft és a Novell vitája

A Novell és a KeyLabs nemrégiben kiadott beszámolója szerint a NetWare 5-ös változata 34 százalékkal jobb teljesítményre képes, mint a Windows NT Server 4.0. A Microsoft szerint ez az állítás félrevezető, mert csak a tesztelési környezet alakította így az eredményt, valós élethelyzetekben végzett mérések szerint a Windows NT Server 4.0 mintegy 26 százalékkal jobb teljesítményt nyújt, mint a NetWare 5. A Novell ugyanis a teszt során a NetWare kiszolgálókat oly módon konfigurálta, ahogy azt a felhasználók nem tennék meg. Az Opportunistic Locking (OL) bekapcsolását korábban a Novell is problematikusnak ítélte, mert adatvesztést és más gondokat okozhat. Nem világos bizonyos esetekben az OL bekapcsolásának módja sem, és nem használható együtt a Novell új tárolórendszerével, az NSS-sel. Ezzel szemben az Opportunistic Locking jellemzőinek megfelelő saját megoldását a Microsoft hosszasan tesztelte, és az már a Windows NT Server megjelenése óta alapértelmezett beállításként használható, a felhasználónál kompatibilitási és adatbiztonsági problémát nem okozva. A Microsoft másik ellenérve, hogy a teljesítmény és ár arányát nézve a Windows NT Server 4.0 még mindig sokkal előnyösebb, mint a NetWare. A Mindcraft céggel elvégeztetett „ellenfelmérés" alapján készült jelentés a http://www.mindcraft.com/whitepapers/nts4nw5filesvr.html címen olvasható.

Norton AntiVirus

A százezres példányszámban megjelenő Secure Computing magazin a hálózatok biztonságáért felelős informatikai szakemberek egyik legfontosabb szakmai fóruma. A laphoz tartozó West Coast Labs elsőként a Norton AntiVirus programnak ítélte a legmagasabb szintű „Checkmark Level Two" minősítést. A Norton AntiVirus többszintű vírusvédelmi termékcsalád, amely az asztali számítógép, a szerverek, a csoport-szerverek, az Internet és a gateway szintjén biztosítja a vírusérzékelést és vírusmentesítést. A vizsgálatok során egyrészt ellenőrizték, hogy a szoftver képes-e a „Joe Well" víruslistán szereplő, azaz a gyakorlatban terjedő és fertőző összes vírus kiirtására, másrészt azt, hogy beilleszthető-e program a vállalatok biztonsági stratégiájába. A teszt megköveteli a terméktől, hogy megszüntesse a vírusfertőzést, a számítógép adatai viszont maradjanak sértetlenek. A tesztelési eljárást legalább háromhavonta megismétlik, és ha a termék nem teljesíti a megkövetelt szintet, akkor a Checkmark minősítését felfüggesztik, sőt, ha a gyártó nem éri el azt újra, akkor vissza is vonják. A Checkmark aktuális állapotát a West Coast Labs weblapján, a www.check-mark.com címen lehet megtekinteni.

Katmai = Pentium III

Az Intel következő processzorgenerációja Pentium III néven jelenik meg. Az eddig Katmai munkanéven szereplő termék Pentium II magra épül, de további 70 utasítást tartalmaz, amelyek felgyorsítják a multimédia teljesítményt. A Pentium III megjelenése 1999 március elejére várható, órajele először 450 és 500 MHz-es, de év végéig el akarják érni a 600 megahertzet. Forgalomba hozatalakor a chip ára 500 dollár felett lesz, de elemzők szerint fél év alatt ez akár 200 dollár körülire is csökkenhet. Mivel az Intel számára a Pentium név igen nagy befektetés volt, és jól bevált, ezért valószínű, hogy a következő generációk is ezt fogják viselni. Az Intel többi márkaneve nem ilyen sikeres, a Celeron és a Xeon elnevezés például sok kritikát kapott.

A Linux térhódítása

A Linux operációs rendszer forgalma 1998-ban 212 százalékkal nőtt, nagyobb ütemben, mint bármelyik más szerverrendszeré. Az IDC becslése szerint ezzel a Linux piaci részesedése 6,8 százalékról 17,2 százalékra ugrott, aminek fő motívuma a Microsoft-ellenesség, továbbá a Linux nagy teljesítménye, alacsony ára, igények szerinti méretezhetősége és szabad forráskódja. A kiadott licencek száma 1998-ban elérte a 4,4 milliót, ami 1997-ben 236 ezer, 1998-ban 748 ezer új linuxos szervert jelent. Az IDC legfrissebb felmérése szerint a szerverek 24 százaléka fut NetWare-en és 36 százaléka Windows NT-n, a Unix 17,4 százalékkal csak hajszálnyira van a Linux előtt. Az NT éveken át nagy ütemben terjedt, ami groteszk módon részben a multitaszk feladatok ellátásában megnyilvánuló gyengeségének volt köszönhető: a felhasználók kénytelenek voltak a feladatokat több, egymástól elszigetelt szerver között megosztani — mondotta Dan Kusnetzky az IDC képviseletében. Miközben Unix szerverek 5–10 ezer e-mail levelezőt rutinból kiszolgálnak, a legtöbb, amit az IDC egy NT-s gépen látott, 3–400 között volt.

Szerverek — hardver oldalról

Ugyancsak az IDC adatai szerint 1998 harmadik negyedévében a 25 ezer dollár alatti szerverek piacán a Compaq a listavezető, 32%-os részesedéssel, ami 3,6%-kal több mint az előző év hasonló időszakában. A 2. helyre a Hewlett-Packard jött fel 15,2 %-os részaránnyal (egy évvel korábban 12,3%), míg az IBM a harmadik helyre esett vissza, egy évvel korábbi 13,1%-os részesedésük 12,9%-ra csökkent. Páratlan mértékű forgalomnövekedésének eredményeként a Dell több helyet ugrott

előre, és most 11,5%-kal a negyedik helyen áll, 1997-ben a részesedése csak 7,9% volt. Ezt a növekedést a Dell annak köszönheti, hogy az USA-ban egyre többen vásárolnak szervereket (és egyéb rendszereket) a cégtől közvetlenül. Az így vásárolt gépek aránya most 19,7%, míg 1997 hasonló időszakában csak 13,4% volt. Az olcsó szerverek piacán a négy első helyezett (Compaq, HP, IBM, Dell) a piac 70%-át tartja kézben. Ebben a piaci szegmensben 1998 harmadik negyedében 22%-kal volt magasabb a forgalom, mint 1997 hasonló időszakában, tehát bővült az olcsó szerverek piaca. Az ötödik helyezett a Gateway (4,3%), a hatodik a Micron (1,8%).



Nagy és kis halak

Ugyancsak az IDC új tanulmánya foglalkozik a Weben történő beruházások várható alakulásával. A hagyományos médiavállalatok, amelyek eddig a partvonal mellől figyelték az eseményeket, 1999-ben a jelek szerint beszállnak a játékba. A behemót médiacégek felvásárolnak kisebb, de neves webeseket. Már lezajlott az America Online tranzakciója (4,3 milliárd dollárt fizetett a Netscape-ért), a Disney az Infoseekbe, az NBC a Snap-be fektetett be (amely közös vállalat a CNET-tel). A tanulmány azt jósolja, hogy 1999-ben a Time Warner vagy a CBS igyekszik megszerezni partnernek a Yahoo-t. A médiabirodalmakon kívül műszaki cégek és pénzügyi szervezetek is megjelennek befektetőként az elektronikus kereskedelemben, így a Microsoft, a Citigroup és a Wells Fargo is szóba jöhet. Három további jóslat: a Compaq eladja az AltaVistát, az Infoseek összeolvad a Lycos-szal, és az NBC növeli részesedését a CNET-ben és a Snap-ben.

„Macintosh Explorer 4.5"

Nem az első eset, hogy a Microsoft először Macintosh platformon mutatja be valamelyik újítását. Most az Internet Explorer 4.5 volt a soron. Egy új webhely, a Mactopia szintén azt támasztja alá, hogy a Microsoft komolyan veszi a Macintosh platfomot. A lényeges változás az eddigiekhez képest, hogy nem mechanikusan teszik át a windowsos programokat Mac-re, hanem kihasználják annak többletlehetőségeit, sajátos funkcióit, adottságait. Az Internet Explorer 4.5 például támogatja a Sherlock nevű Apple programot, amely megkönnyíti a tartalom szerinti keresést a Weben, és több más sajátossággal is rendelkezik. Ez az akció a Microsoft és az Apple közötti áttekinthetetlen kapcsolat egyik látványos megnyilvánulása, amelynek mozgatórugója a Microsoft részéről nyilvánvalóan a trösztellenes törvény alapján ellene indítható eljárás kivédése.

Konfliktus a 3Com-nál

1998. december elsejével felmondott a 3Com magyarországi irodájának eddigi vezetősége. Batizfalvy Tamás — aki 1996. márciusa óta vezette a 3Com hazai képviseletét — úgy látta, hogy az iroda kiemelkedő értékesítési teljesítménye ellenére nem kapott munkájához megfelelő támogatást (például létszámbővítést) és fokozatosan ellehetetlenült. Sajnos nem az első eset, hogy a külföldi tulajdonosok és a magyar képviselet között nincs megfelelő összhang, és eddigi tapasztalataink szerint eddig többnyire nem a határ innenső oldalán volt a hiba oka.

Nem csak OS/2 ...

Java 1.1.7

Az IBM kibocsátotta a legújabb, 1.1.7-es sorszámot viselő Java motort OS/2-re. Aki rendszeresen használ Java nyelven írt alkalmazásokat, vagy fejleszt Java alá, annak mindenképpen érdemes beszereznie a frissítést, mert annak teljesítménye jóval nagyobb. Az IBM saját mérései szerint (http://www.software.ibm.com/os/warp/performance/jdk117_os2perf.htm) az OS/2-es Java motor teljesítménye már majdnem kétszerese a Windows NT 4.0-n futtatott Microsoft Java SDK 2.0-nak. A termék megtalálható az IBM Software Choice oldalán (http://service.software.ibm.com/asd-bin/doc/index.htm). A Java programok fejlesztése iránt érdeklődőknek érdemes lehet ellátogatni az Innoval cég kiszolgálójára is (http:/www.innoval.com), ahol ezentúl ingyenesen adják a J Street Mailer Java alapú levelezőprogram vásárlóinak a forráskódot is.

FTP fájlrendszer

Nagyon érdekes fájlrendszer fejlesztésébe kezdett Vit Timchishin (tvv@sbs.kiev.ua), ukrán programozó. Az FtpIFS segítségével ugyanis meghajtóbetűket lehet OS/2 alatt ftp kiszolgálókhoz rendelni, így megszűnnek a hagyományos ftp kliensek jellegéből adódó kényelmetlenségek. A hozzárendelés megteremtése után gyakorlatilag bármelyik fájlmenedzser, biztonságimásolat-készítő program vagy saját gyártmányú kötegelt állomány használható lesz ftp-vel. Az új fájlrendszernek rendszeresen jelennek meg újabb verziói, ezért érdemes gyakran figyelni, hogy mikor tűnik fel a http://hobbes.nmsu.edu/incoming könyvtárban az ftpifs*.zip legújabb kiadása.

XFree86 3.3.3

Holger Veit kibocsátotta az ingyenes ablakkezelő rendszer, az XFree86 OS/2-esen futó 3.3.3 változatát. Az XFree86 eredetileg a Unix rendszerek egyik ingyenes ablakkezelője volt, amelyet néhány évvel ezelőtt vittek át más rendszerekre, többek között OS/2-re is. Az XFree86 segítségével viszonylag könnyen átvihetők más unixos alkalmazások is OS/2-re (lásd például múltkori hírünket a Gimpről). Az új XFree86 főleg javításokat és támogatást tartalmaz a legújabb videokártyák számára. A meglehetősen terjedelmes alkalmazás a http://borneo.gmd.de/~veit/os2/xf86os2.html oldalon található.



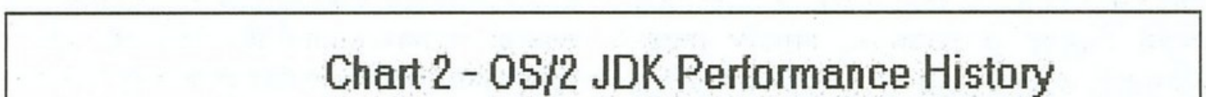
AOL Instant Messenger

InJoy tűzfal kábelmodemekhez

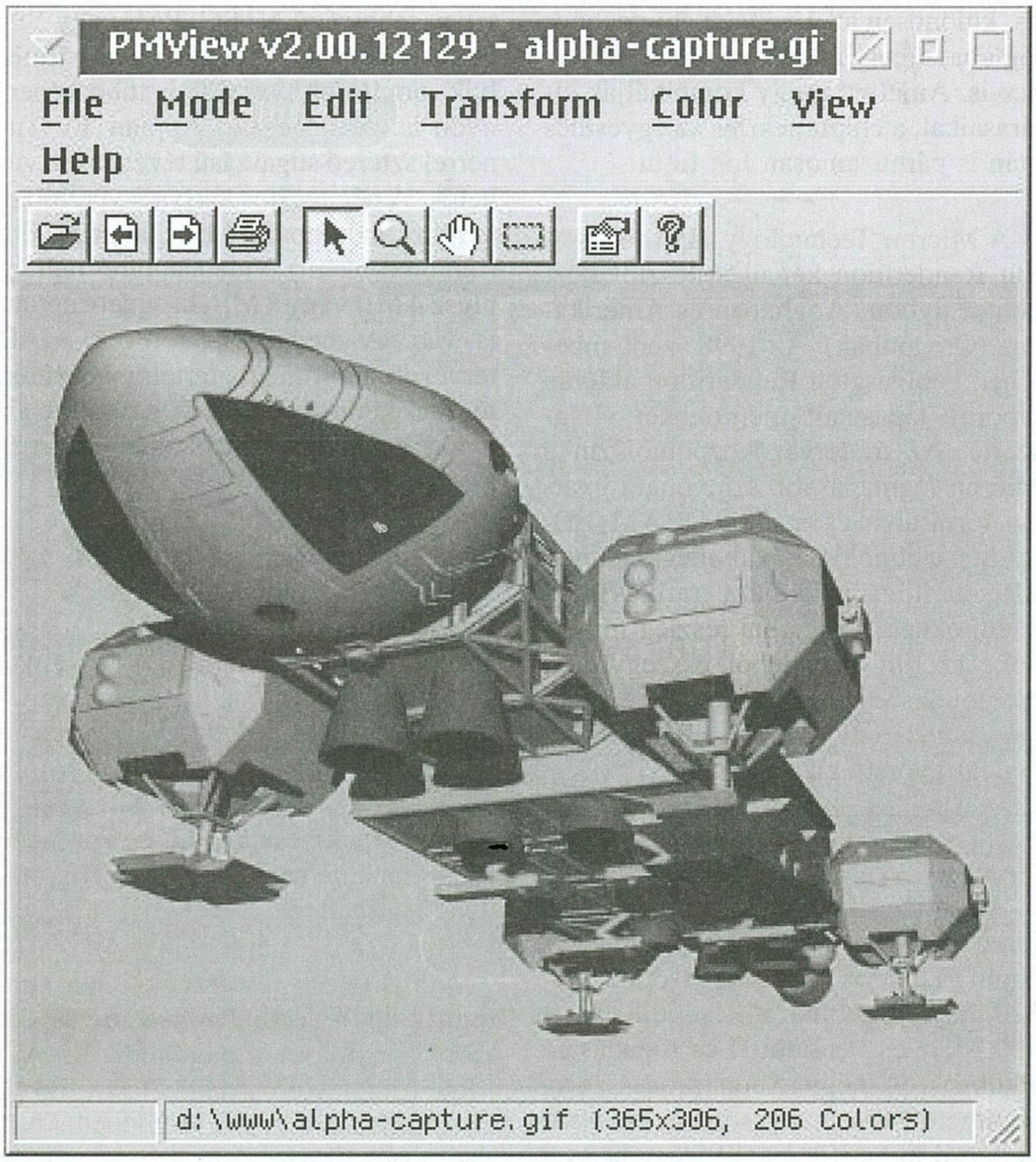
Nagyon sok OS/2 felhasználó számára ismerősen cseng az InJoy tárcsázóprogram neve, amelyet a dán illetőségű F/X Communications cég fejleszt. A terméknek már eddig is több változata létezett, most pedig tovább bővült a választék a kábelmodemeket támogató tűzfal (InJoy Firewall 1.0) kibocsátásával. A cég tervei szerint ez a termék lesz az alapja a közeljövőben kifejlesztendő tűzfal eszközkészletnek, amelynek segítségével technikailag nagyon könnyen megvalósítható lesz, hogy kábelmodemmel az Internetre kapcsolódó OS/2-es gépen keresztül egy egész intranetet biztonságos Internet kapcsolattal lássanak el. Az érdeklődők további információhoz juthatnak a cég honlapján: http://www.fx.dk/firewall.

BootOS2 9.17

Megjelent a hobbes-on a BootOS2 legújabb frissítése (http://hobbes.nmsu.edu/pub/os2/util/system/btos2917.zip, amelynek segítségével nagyon egyszerűen és gyorsan lehet indítható OS/2-es rendszereket készíteni merevlemezre vagy floppyra, már installált OS/2 2.x, 3.x vagy 4.0 rendszerek alapján. A telepítendő rendszer szolgáltatásait (PM, WPS, REXX stb.) paraméterekkel



Java motorok versengése



Előkészületben a PMView 2.0

adhatjuk meg, és Warp 3-as alapon, kb. 10 MB-on már gyakorlatilag teljes funkcionalitású rendszert lehet kialakítani, amely kitűnően használható a fő rendszer karbantartására. „Mini” alapváltozatban a rendszer egyetlen floppyra ráfér, mégis jól használható.

IBM-es javítások

Megjelentek a menetrendszerű javítások az IBM-es termékekhez. A legújabb Warp 3-as, 39-es amerikai javítócsomag az ftp://service.boulder.ibm.com/ps/products/os2/fixes/v3.0warp/english-us/xr_w039/ címen található. A korábbi híreinkben említett 8610-es MPTS-hez is megjelent az első javítás, amely a SLIP támogatás hibáját javítja (ftp://ftp.boulder.ibm.com/ps/products/tcpip/fixes/v4.2os2/ic22633/).

PMView 2.0 béta

Az OS/2-es körökben igen népszerű képnézegető programnak, a PMViewnak rövidesen megjelenik a 2.0-s, már Windows NT-n és Win95/98-on is futó kiadása. Regisztrált felhasználók a http://www.pmview.com/us/future.html oldalon keresztül jelentkezhetnek a bétaprogramért. A PMView legutolsó, első generációs változata az 1.03, amely szintén nemrég jelent meg a fenti honlapon. Ez az utolsó kiadás, amelynek megvásárlása esetén még egy életre szóló licencjogot kapunk az OS/2-es kiadásokra. A frissítés főleg fájlformátumokkal (TGA, IFF, TIFF) és képlementési funkciókkal kapcsolatos javításokat tartalmaz.

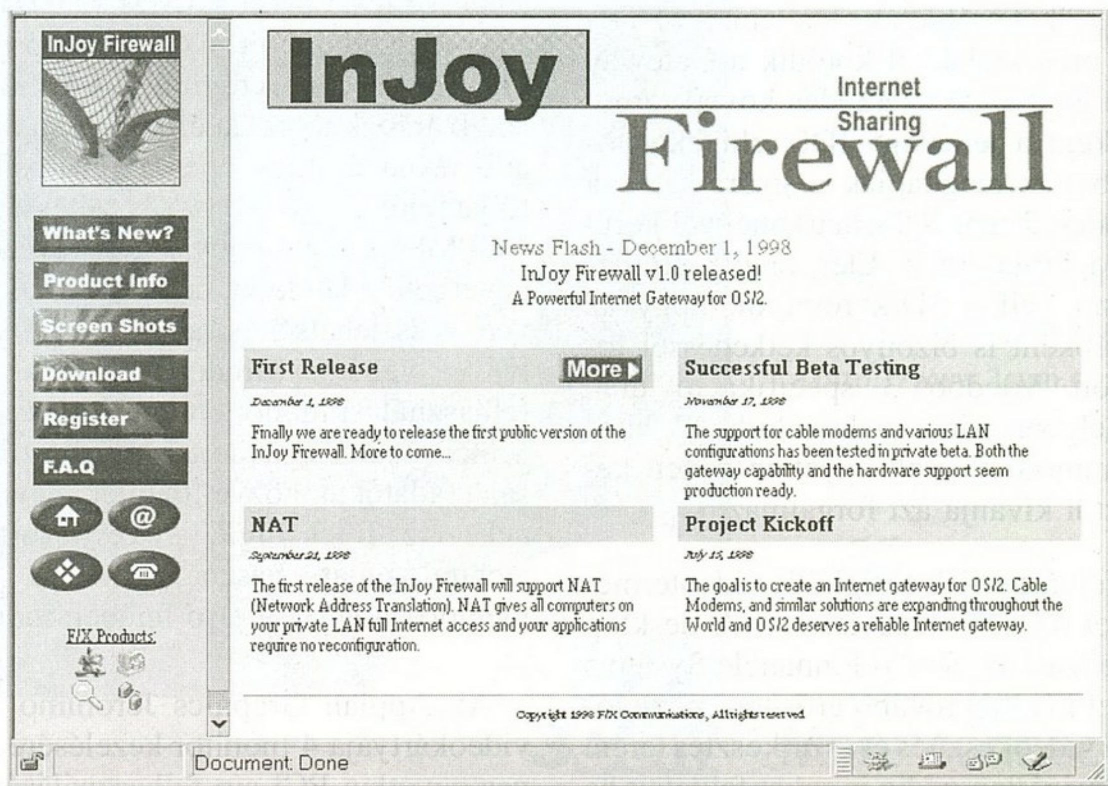
AOL Instant Messenger

Sokáig nem volt elérhető OS/2-re az AOL Instant Messengere, amelynek segítségével könnyebben érintkezhetünk a hálózaton régi és új ismerőseinkkel. Most megtört a jég, mert az AOL kibocsátotta az alkalmazás Java verzióját, amely a hírek szerint tökéletesen működik az új 1.1.7-es OS/2-es Java motorral. Az Instant Messenger letölthető a http://www.aim.aol.com/beta/javadownload.html oldalról.

FaxWorks LAN verzió

A Keller Group piacra dobta az OS/2 Warp BonusPak-ből is ismert FaxWorks hálózati megosztással is használható változatát. A termék segítségével könnyedén megoldható egy kisebb iroda számítógépeinek faxosítása. A FaxWorks-öt a kiszolgálóra telepítve ugyanis öt másik, természetesen hálózati összeköttetéssel rendelkező munkaállomás használhatja a kiszolgáló faxát. Ezzel nemcsak azt érjük el, hogy kevesebb faxmodemre van szükség, hanem azt is, hogy egyetlen telefonvonallal megoldjuk öt munkaállomás ellátását. A terméknek van a Windowson futó változata is, így vegyes környezetben is kitűnően megállja a helyét. Az új FaxWorks megrendelhető a fejlesztő cég honlapján keresztül: http://www.kellergroup.com.

Kádár Zsolt



Az Injoy tanyája

Kissé rendhagyó módon az első néhány hír nem konkrét hardver eszközökről szól, hanem olyan üzleti lépésekről, melyek általában a háttérben maradnak, pedig jelentősen befolyásolják az új termékek megjelenését.

Szerződések, kapcsolatok

10 évre szóló kölcsönös licencmegállapodást kötött egymással az Intel és az S3. A fejlesztői szabadalmak átadása vonatkozik bizonyos félvezető termékekre, az Intel jelenlegi és jövőbeni általános felhasználású processzorbuszaira stb. Az Intel emellett S3 részvényeket szándékozik vásárolni. Az Intel AGP 4x-es rendszerének az S3 lesz az ellenőrző partnere, így jó helyzetbe kerül ahhoz, hogy elsőként bocsásson ki AGP 4x-nek megfelelő termékeket, melyek az előrejelzések szerint 1999 második felében az adott termékcsoport több mint 40%-át teszik majd ki.

>>

A grafikus chipgyártók közül elsőként az S3 készített 0,18 mikronos chipet. Az UMC eddig már három 0,25 mikronos megoldást dolgozott ki az S3-nak, míg a többi gyártó ebben a kategóriában még csak most próbálja ki első modelljeit.

>>

141 millió dolláros tőzsdei akció keretében a 3Dfx megvásárolja az STB Systems céget. Valamennyi STB részvényes 0,65 3Dfx részvényt kap minden egyes STB részvény után. Az üzlet révén a 3Dfx közvetlen kapcsolatba kerülhet a felhasználókkal, akikhez eddig csak a kártyagyártók (több mint 30 cég) közvetítésével juthatott el, ők viszont a 3Dfx helyett inkább saját márkanevüket próbálták erősíteni. Az STB-hez erős szálakkal kötődik a Gateway és a Dell is. A szerződés következményeként a jelenlegi 3Dfx chipeket továbbra is megkapják a gyártók, de a Voodoo 3 már STB márkanévvel kerül majd forgalomba. Elég nagy optimizmusra vall a 3Dfx részéről, hogy az egyébként is bizonyos kétkedéssel fogadott Voodoo 3 specifikáció után (amelyben nincsenek valódi 32 bites üzemmódok) csak egyetlen cégen keresztül kívánja azt forgalmazni.

>>

Felvásárolják a digitális videotermékeket készítő Truevision-t. A desktop videózásban vezető Pinnancle Systems ezzel kívánja tovább erősíteni pozícióját a videokészítés és -szerkesztés terén. A Truevision eddig is sokat fektetett be a következő generációs architekturákba, különösen az új egyedi chiptechnológiába, amely alkalmazható a HDTV-hez is. Amellett, hogy kombinálják eljárásaikat, a chipfejlesztés az egyesülés után is párhuzamosan fog futni.

>>

A Micron Technology által felvásárolt Renderition két új fejlesztői központot nyitott, Angliában és Amerikában (Oregonban). Az 1998 szeptemberében beolvasztott Renderition aktívan toboroz tapasztalt mérnököket világszerte. Az új tervezőközpontokban a Micron legmagasabb színvonalú gyártási forrásaival és integrált DRAM (ED-RAM) technológiájával megerősítheti a Renderition pozícióját a grafikus megjelenítőknél. Az angliai részleg főmérnöke szerint a világból összegyűjtött mérnökök tehetsége révén a cég egy nagyságrenddel túlszárnyalhatja a jelenlegi gyorsítókat.

Munkaállomások

Az Silicon Graphics bemutatta első Intel és Windows NT alapú munkaállomásait. Az új Visual Workstation 320 ára 4000 dollárral kezdődik és két 400 MHz-es Pentium II-es fogadására alkalmas. A négy Xeon processzorig bővíthető 540-es csúcssorozat ára 6000 dollártól indul. (A már 1998-ra beígért modellek csak mostanában várhatók, mert az SGI a Xeon működtetésekor eleinte nehézségekbe ütközött.)

Video, multimédia

WinTV-D néven digitális tévésugárzás vételére alkalmas kártyát készít a Hauppauge Computer Works. A kártya lehetővé teszi a DTV (SDTV, HDTV) adás megjelenítését a számítógép monitorán, a speciális digitális vevőkészülékek árának töredékéért. Mivel a digitális tévéadás nagysebességű soros bitfolyamból áll (6 MHz sávszélességen 19,4 Mbit/s adatmennyiség), a tévéadás keverhető adatokkal, de tiszta adatközvetítés is lehetséges, ami ezzel a kártyával vehető. Mindez várhatóan új felhasználási módot eredményez és piacokat nyit meg, mert a műsorszórók saját adatokat közvetíthetnek vevőiknek. Az Intel digitális vevőszoftver technológiáját használó kártya tömeggyártása 1999 első felében indul.

>>

Az Appian Graphics Jeronimo Pro videokártyája 4 monitor kezelésére képes egyetlen PCI sín felhasználásával. A 4 Permedia chipet alkalmazó kártya teljes sebességű NTSC, PAL vagy SECAM videoót tud megjeleníteni bármelyik monitoron, vagy akár több képernyőn is. A kiegészítő Appian TV Tunerrel sztereó sugárzású tévéadások vehetők. A négyportos kártya 3 verzióban kerül forgalomba: 4 MB csatornánként (monitoronként) videocapture nélkül, plusz 4 MB vagy 8 MB, ha videocapture is van. A megcélzott rétegek: CAD tervezők, multimédia tartalomkészítők, tőzsdei kereskedők (akik egyszerre figyelik a piaci változásokat, a híreket és a részvényárakat).

>>

A Metabyte bejelentette, több mint 200 játék támogatja a Wicked3D EyeScream vezeték nélküli sztereoszkópikus megjelenítőjét. A Metabyte eddig is a kártyáihoz adott szoftvertámogatással emelkedett ki versenytársai közül, például az egyedileg módosított 3Dfx driverekkel, amelyek a meghajtóprogramon keresztül működő minden programnál lehetővé teszik a 3D szemüveg használatát.

>>

A Macintosh felhasználók sem maradnak ki a legújabb generációs 3D kártyák nyújtotta előnyökből. Az ATI Rage 128GL sorozatának PC-s változatait Apple környezetre is adaptálta Nexus 128, Xclaim VR 128 és Orion fantázianevekkel.

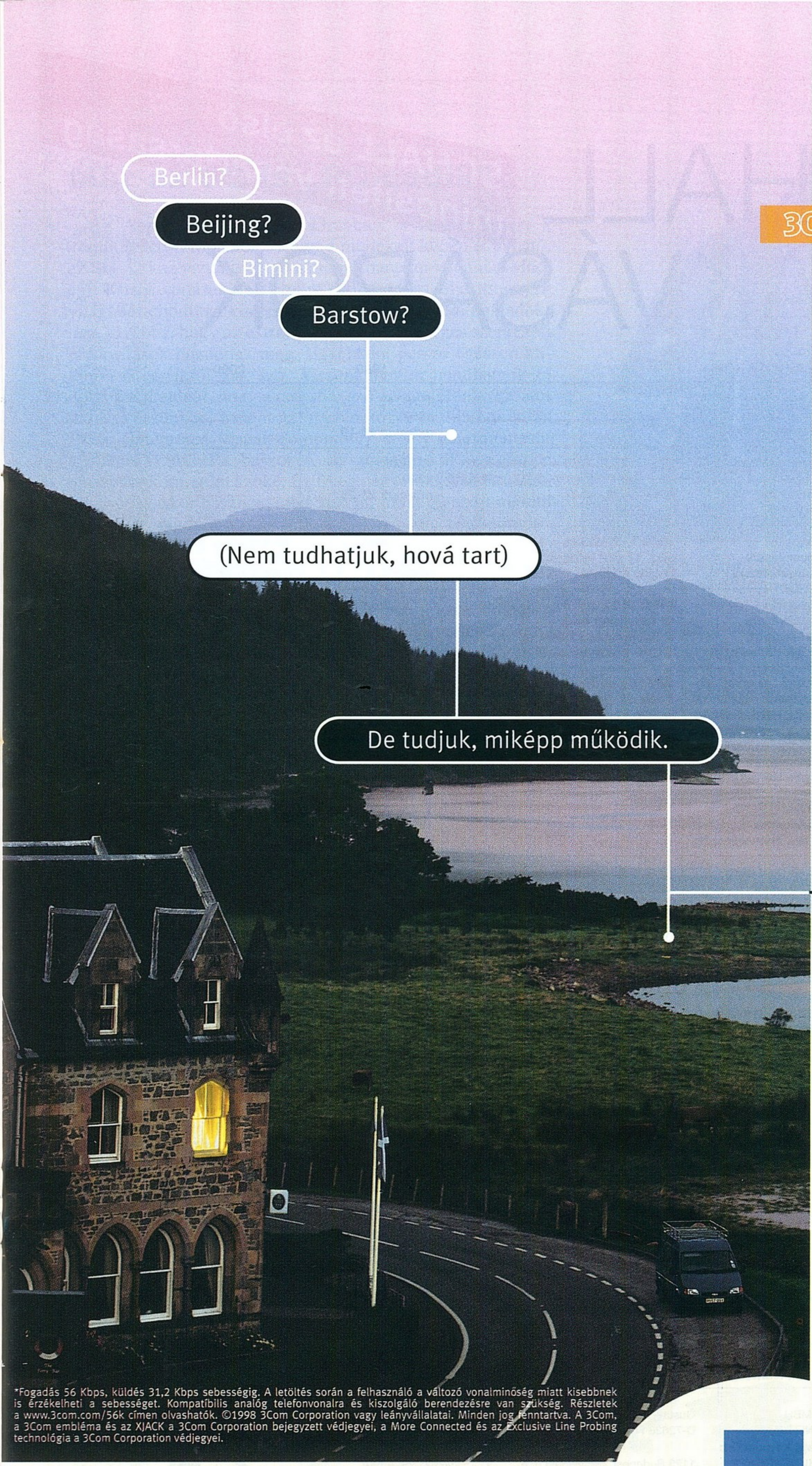
Szoftver, driver

Az első DVD játék tavaszra várható. A Lander (Psygnosis) MPEG-2 full-motion videorészleteket, Dolby Digital 5.1 csatornás zenét és hangeffekteket tartalmaz, valamint valódi interaktív élményt nyújt látványban és hangban. Az űrben játszódó kaland „példaértékűen mutatja be a szórakoztatóiparban jelenleg használt fejlett technológiát" — állítja Harry Vitelli a cég marketing alelnöke. A lemez úgy készült, hogy normál (nem számítógépes) DVD lejátszóval is megtekinthetők a videobejátszások, vagy lejátszhatók a zenesávok.

>>

A Miles Audio System 5.0 beépíti Creative EAX 2.0-s 3D modelljét a támogatott hangrendszerek közé. A Miles Audio — a Smacker Video és a Bink Video mellett — már 1991 óta része a RAD Games Tools játékfejlesztői környezetnek. Ezeket a rendszereket 1994-től 1800-nál is több játéknál alkalmazták és 1998-ban 35 millió játékot adtak el velük.

Bánó György



*Fogadás 56 Kbps, küldés 31,2 Kbps sebességig. A letöltés során a felhasználó a változó vonalminőség miatt kisebbnek is érzékelheti a sebességet. Kompatíbilis analóg telefonvonalra és kiszolgáló berendezésre van szükség. Részletek a www.3com.com/56k címen olvashatók. ©1998 3Com Corporation vagy leányvállalatai. Minden jog fenntartva. A 3Com, a 3Com embléma és az XJACK a 3Com Corporation bejegyzett védjegyei, a More Connected és az Exclusive Line Probing technológia a 3Com Corporation védjegyei.

Úton vannak. Levegőben vannak. Kiejthetetlen nevű helyeken fordulnak meg. De ha 3Com® Megahertz® PC

3Com Megahertz

kártyát használnak, akkor biztosak lehetünk afelől, hogy utazó kollégáink bárhonnan, bármikor képesek a kapcsolódásra.

Akár LAN-on, WAN-on, ISDN-en vagy GSM-en használják, a 3Com Megahertz PC-kártyáknál nincs megbízhatóbb. A vonalhibák ellensúlyozásáról, így a jobb teljesítményről a kártyába épített saját vonalvizsgáló (Exclusive Line Probing) technológiánk gondoskodik; és az egyedülálló XJACK® csatlakozónak köszönhetően az utazónak nincs szüksége saját kábelre, amelyet könnyen elveszíthet, amelyet cipelnie vagy pótolnia kell.



3Com Megahertz PC Cards

- ✓ Exclusive Line Probing technológia
- ✓ Egyedülálló XJACK® csatlakozó
- ✓ Gyors, 56K* sebességű kapcsolódás szabványos telefonvonalon
- ✓ Garantált kompatibilitás az ipari szabványokkal

Többet is megtudhat a 3Com Megahertz termékeiről és szolgáltatásairól, ha megtekinti honlapunkat a http://www.3com.com címen.

3Com® More connected.™

SCHALL az első minősített vásártársaság

SCHALL SZAKVÁSÁROK

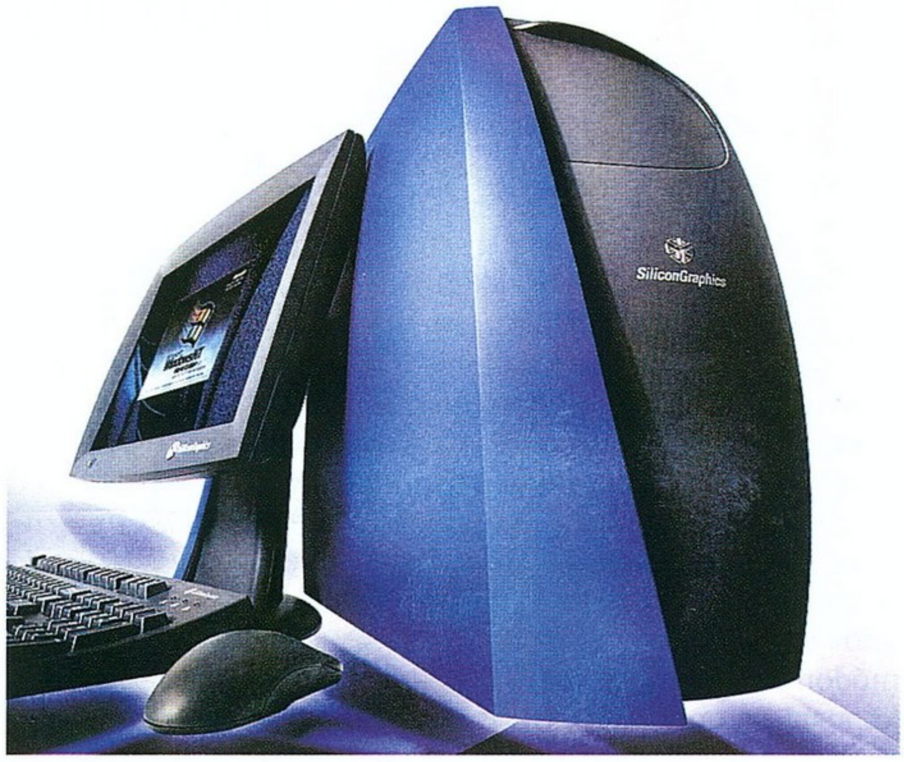
SCHALL SZAKVÁSÁROK		1999	2000
Az autófejlesztés szakvására **Sinsheim, Németország**	—	1. A•T•E **1999. március 16-19.**	—
Szerelési- és karbantartás-technikai nemzetközi szakvásár **Westfalenhallen Dortmund, Németország**	**1997: 113 kiállító + 25 képviselt cég, 4 011 látogató, 5 200 m² brutto kiállítási terület**	7. Techmo **1999. március 23-26.**	—
A minőségbiztosítás nemzetközi szakvására **Sinsheim, Németország**	**1998: 622 kiállító + 44 képviselt cég, 17 210 látogató 26 500 m² brutto kiállítási terület**	13. Control **1999. május 4-7.**	14. Control **2000. május 9-12.**
A minőségbiztosítás nemzetközi szakvására **Elmia Városközpont, Jönköping, Svédország**	**1998: 110 kiállító + 43 képviselt cég, 6 232 látogató 2 900 m² brutto kiállítási terület**	2. Qualitec **1999. május 18-20.**	—
Gyártástechnológia az orvostechnikában - nemzetközi szakvásár **Frankfurt/Main, Németország**	**1998: 54 kiállító + 2 képviselt cég, 6 350 látogató, 1 500 m² brutto kiállítási terület**	2. TechMed **1999. június 9-11.**	—
Steril technológiák és eszközök nemzetközi szakvására **Frankfurt/Main, Németország**	**1998: 126 kiállító + 10 képviselt cég, 6 350 látogató, 2 500 m² brutto kiállítási terület**	2. CleanRooms Europe **1999. június 9-11.**	—
Tisztítástechnikai nemzetközi szakvásár **Frankfurt/Main, Németország**	—	1. CleanTech **1999. június 9-11.**	—
Szerelési- és karbantartási nemzetközi szakvásár **Sinsheim, Németország**	**1998: 632 kiállító + 74 képviselt cég, 25 160 látogató, 29 500 m² brutto kiállítási terület**	18. Motek **1999. szeptember 21-24.**	19. Motek **2000. szeptember 19-22.**
A grafikai ipar szakvására **Sinsheim, Németország**	**1997: 98 kiállító + 5 képviselt cég, 4021 látogató, 2300 m² brutto kiállítási terület**	5. Druck + Form **1999. október 6-9.**	—
Szerszám és szerszámtechnikai szakvásár **Sinsheim, Németország**	—	1. EuroTools **1999 október 6-9.**	—
A műanyagfeldolgozás nemzetközi szakvására **Friedrichshafen, Németország**	**1997: 927 kiállító + 106 képviselt cég, 31 953 látogató, 50 000 m² brutto kiállítási terület**	13 Fakuma **1999. október 19-23.**	14.Fakuma **2000. október 17-21.**
Szennyvíz- és víztechnológiai szakvásár **Sinsheim, Németország**	**1998-ban az AWL Tech és a Südblech kettős rendezvény volt: 201 kiállító + 9 képviselt cég, 5147 látogató, 10 000 m² brutto kiállítási terület**	—	5. AWL Tech **2000. március 29-31.**
Lemezmegmunkálási és illesztéstechnikai szakvásár **Sinsheim, Németország**		—	4. Südblech **2000. március 29- április 1.**
Fémmegmunkálási nemzetközi szakvásár **Nürnberg, Németország**	**1998: 539 kiállító + 94 képviselt cég, 29107 látogató, 25 200 m² brutto kiállítási terület**	—	17.Fameta **2000. május 2-6.**
Optikai, optoelektronikai nemzetközi szakvásár, Felhasználás és technológia **Frankfurt/Main, Németország**	**1998: 308 kiállító + 142 képviselt cég, 6350 látogató, 10 541 m² brutto kiállítási terület**	—	5.Optatec **2000. június 27-30.**

Szervező:
P.E. SCHALL GMBH MESSEUNTERNEHMEN
Gustav-Werner-Strasse 6 D-72636 Frickenhausen
Tel: 00 49/70 25/92 06-0 Fax: 00 49/70 25/92 06-20

Magyarországi képviselet:
OPEN GATES HUNGARY KFT. Tel: 214-9943, 214-9059
1123 Budapest, Nagyenyed u. 6. E-mail: OGH. Kft @mail.datanet.hu
Tel/Fax: 3562-890

INFORMÁCIÓKÉRÉS: 17 ▲

Már az SG is „PC-sedik”



Az Apple után a Silicon Graphics is kedvet kapott a gömbölyded formákhoz. Amint a mellékelt képen is látható, az SG 320-as vizuális munkaállomásban van valami „iMac-es”. Közös vonásuk az is, hogy a különleges teljesítmények világából elindulva közelítenek az egyszerű földi halandókhoz, továbbá, hogy az esztétikum mögött lényeges szerkezeti módosítások húzódnak meg. Egyébként persze egészen különböző gépekről van szó. Viszont megint tanulhatunk új dolgokat. Például, hogy a crossbar szó ne a telefonközpontot juttassa eszünkbe, hanem az új számítógép-architektúrát, az IVC-t (Integrated Visual Computing), amely tehermentesíti a PCI és az AG2PX buszokat, sávszélessége tízszer, illetve hatszor akkora, mint azoké. Ez főleg nagy modellek és képek feldolgozásakor hasznos, mert kezelni tudja gigabájt méretű képeinket, digitális videoszerkesztéskor pedig négy tömörítetlen videofolyam teljes sávszélességét.

Maximális kiépítettség esetén két Pentium II processzor, 1 GB RAM, 2 harddiszk és 3 PCI bővítőhely lehet benne (nagyobbik testvérében, az 540-esben pedig ezek duplája). A grafika 1920x1200-as felbontású. Alapfelszereléséhez tartozik mindenféle ki- meg bemenet: 10/100-as Ethernet, IEEE-1394, USB, analóg video és audió, CD... sőt még floppymeghajtó is van benne (ellentétben az iMac-kel!). A Silicon Graphics-nak ezek az első gépei, amelyek Intel processzorra és Windows NT operációs rendszerre épülnek. Bármily meglepő, a teljes kiépítésű Silicon Graphics 320-as ára 1 millió forint alatt van.

Ipar és művészet

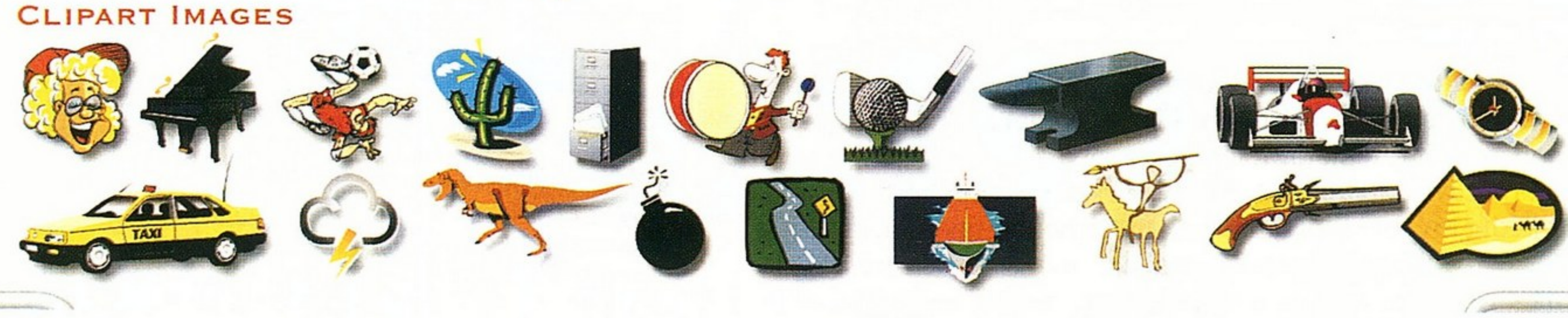
Amióta a technika lehetővé tette, hogy a grafikai (művészi) munkák meggyorsítására sablonokat használjanak, felemás viszonyban vagyunk az ilyen segédeszközökkel. A Letraset matricaszerűen átmásolható betűi és kisebb rajzai nálunk is a grafikusok kelléktárába tartoztak, jóval a számítástechnikai korszak előtt. Az is igaz viszont, hogy Magyarországon az előre gyártott rajzok és grafikák soha nem voltak népszerűek, a grafikusok inkább csak a betűket és a geometriai ábrákat használták fel. Grafikai alkotást matricával összehozni nálunk valahogy nem fért össze az egyediség igényével (vagy a hiúságunkkal?).

A számítógép azután lehetővé tette, hogy igen könnyen és elképesztő mennyiségben kerüljenek „mélyhűtött” nyersanyagok a grafikai hajlammal megáldott dilettánsok képernyőjére — és onnan mindenféle felületre. A legújabb ilyen gyűjtemény a Corel képgalériája, 1 millió darab illusztrációval: fotóval, rajzzal, mintázattal stb. A dolog elgondolkodtató, és érdekes párhuzamba állítható a divattal. Valamikor a nőknek traumát okozott, ha az utcán felbukkant mellettük valaki teljesen ugyanolyan ruhában. Még kínosabb volt a helyzet, ha ez egy házibulin vagy fogadáson fordult elő. Később „a hozzáállás” kicsit változott, az egyenruha a kasztosodás öntudatstimuláló eszköze lett, a farmerek műkoptatásában és műszaggatásában éppúgy, mint az elegáns kosztümök színében és szabásában. Manapság viszont „minden virág virágzik”, és a két véglet között nyugodtan ingázhat az, akinek van hozzá kedve és ideje... no meg pénze.

Mi lesz tehát a 14 CD-n szállított 1 millió darab illusztráció sorsa? Szembetalálkozunk-e velük, és kuncogunk-e, hogy valaki ihlet helyett megint ezt a nem túl eredeti megoldást választotta? De az 1 millióból majd 10 millió és 100 millió lesz, és egyre csökken annak valószínűsége, hogy bizonyos motívumot keresve ketten ugyanarra a képfájlra kattintsanak, és akkor műalkotásunk már akár eredetinek is tűnhet. Továbbá azt el kell ismerni, hogy a rajzokat és képeket hozzáértők készítették és válogatták. Nem is maguk az alkotások váltják ki ellentmondásos érzelmeinket, hanem a „használatba vétel” körülményei. De, mint tudjuk, a mennyiség egy bizonyos ponton minőségbe csap át. Csak győzzük kivárni.

Az euro kinyomtatása

Az euro bevezetésének kezelésére a nagy multinacionális cégek már 1998 közepe óta „csiszolják” munkafolyamataikat



és szoftvereiket. Az euro szimbólumát természetesen egyre gyakrabban ki is kell majd nyomtatni, és ez a feladat kicsit bonyolultabb, mint amilyennek tűnik. A Hewlett-Packardnál az első lépés a szimbólum beillesztése volt a HP betűkészleteibe. Miután rendelkezésre állt az euro kód definíciója, magát a szimbólumot is elérhetővé kellett tenni a nyomtató számára. Kétféle betűkészlettel lehet nyomtatni. (1) A rendszer betűtípusok általában eredetileg is szerepelnek az operációs rendszerben, de később is felvehetők oda. (2) A másik mód a nyomtatón lévő betűtípusok használata.

Ha a rendszert alapértelmezés szerint a nyomtató betűkészletére állítjuk be, akkor a nyomtató gyorsabban működik, és csökken a hálózati forgalom. A legtöbb HP LaserJet nyomtató alapértelmezésben saját betűtípussal dolgozik, ilyenkor azonban nehéz újabb elemeket beilleszteni a nyomtató betűkészletébe, mert az új kódokat az eredetihez hasonló módon kell betáplálni. (A rendszer betűtípusoknak mindenképpen az operációs rendszer rendelkezésére kell állniuk, hogy láthassuk azokat a képernyőn, és le tudjuk tölteni a nyomtatóra.)

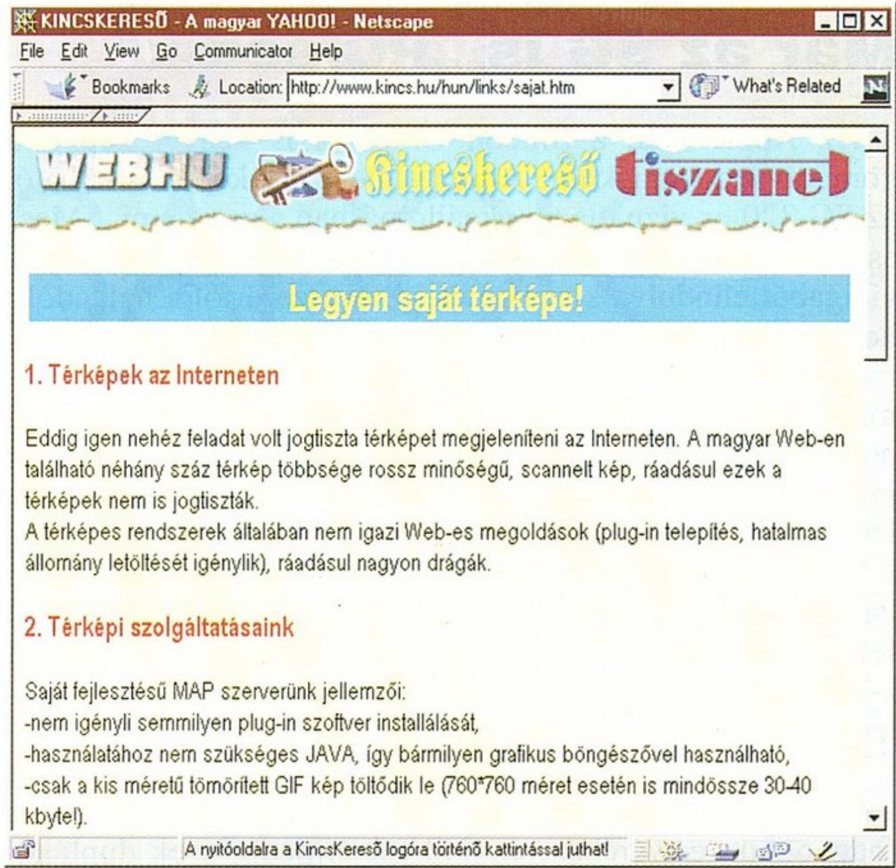
A Windows 95, 98 és az NT 4.0 és 5.0 platformjához adott kiegészítés (lásd CD-mellékletünket) lehetővé teszi az euro szimbólumának megjelenítését a képernyőn, és kinyomtatását is. A Hewlett-Packard frissített meghajtókat és FontSmart betűkészleteket kínál a meglévő szoftverek kiegészítéseként. Ez a beillesztés már megtörtént a HP145 nyomtató betűkészletébe a FontSmart 3.0 verzióval (Win 95, Win 98, NT4 és NT5 felületre).

A HP betűkészletekkel, nyomtatókkal és alapszoftverekkel foglalkozó munkacsoportjai közösen láttak munkához, hogy az euro szimbóluma szerepeljen az 1998 ősze után bemutatott legtöbb új HP nyomtató saját betűkészletében. Azokban az esetekben, amikor a szoftver nem elegendő a felhasználó számára, a HP hardveres megoldást kínál. Olyan régebbi nyomtatók, amelyekben van helye a SIMM vagy DIMM kártyának (a HP LaserJet 4, 5, 6, 4000, 5000, 8000, illetve a színes nyomtatócsalád legtöbb tagja), átállíthatók az euro szimbólum nyomtatására.

Ha olyan régebbi HP LaserJet nyomtatókról van szó, amelyek nem rendelkeznek az említett hardverrel (például az összes kártyahely foglalt) vagy a felhasználó a szoftveres megoldást részesíti előnyben, akkor olyan betűkészleteket kell beszerezni, amelyekben már szerepel az euro szimbólum. A TrueType formátumban készült karakterek felhasználhatók minden windowsos alkalmazásban. Az „eurobetűk" hamarosan elérhetők lesznek a www.hp.com/go/laserjeteuro címen is.

A magyar Yahoo!

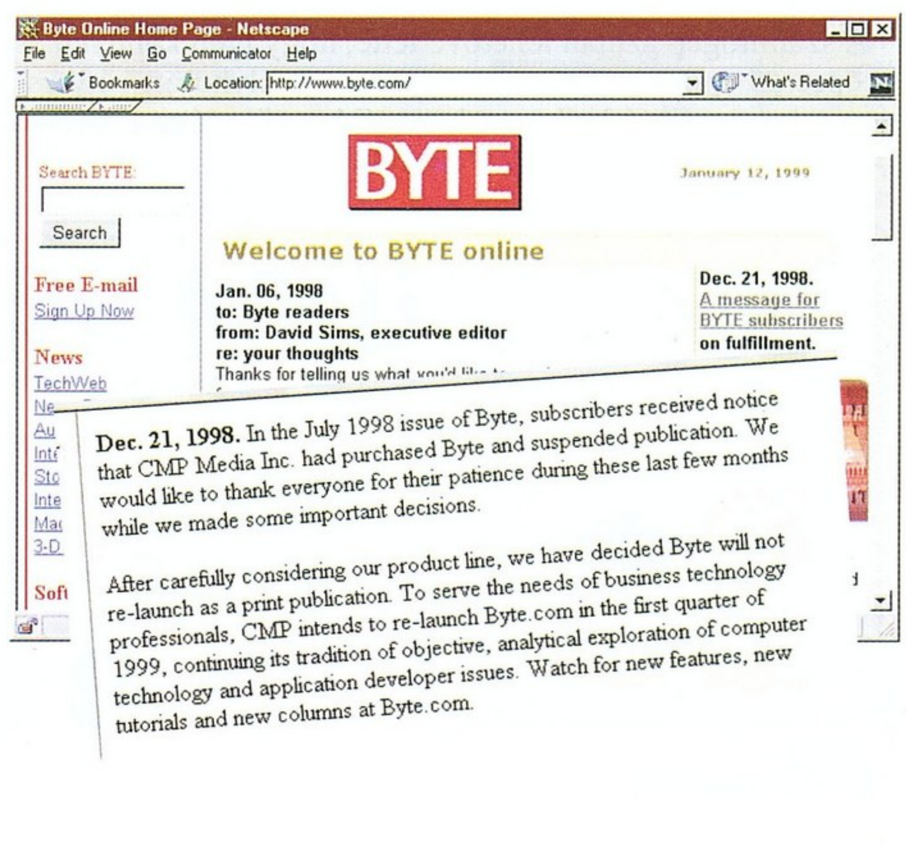
Akinek a figyelmét eddig elkerülte volna, a szókereső rendszerek mellett egy Yahoo! típusú tematikus keresőrendszer is működik a magyar Interneten. A CD-mellékletünkön offline változatban rendszeresen közreadott KincsKereső újdonsága, hogy a webkeresőt összekapcsolták egy térképpel, így ez lett a legnagyobb magyar tematikus webes és térképes keresőszolgáltatás (10 000 térképhivatkozás, naponta 1000–1500 térkép letöltése összesen 12 000 oldalon). A KincsKeresőben szóra, témára, településre egyszerre kereshetünk (450 000 szavas adatbázis), Magyarország és Budapest térképén pedig további szolgáltatások is elérhetők (útvonal, cím, település, szálláshely, mozi, színház, múzeum keresése). Érdemes regisztráltatni is a weblapokat, mert a webhelytu-



lajdonosok a rájuk vonatkozó térképrészletet jogtisztán beépíthetik saját lapjukba. A KincsKereső címe: www.kincs.hu

Reinkarnáció

Kedves ismerőstől fájdalmas a végső búcsú, még akkor is, ha sikerül velünk elhitetni, hogy van feltámadás. A Byte kimúlt, mert kimúlasztották. Amikor 1998 nyarán a CMP megvette és rögtön bejelentette, hogy őszig szünetelteti a lap megjelentetését, sokan már sejtették, hogy ez csak a halálos ítélet végrehajtásának rituáléja. A mellékelt illusztrációk mutatják, hogy a feltámasztási szertartás ennek ellenére zavartalanul folyt tovább, csak most már áthangolva elektronikus médiaként történő reinkarnációra. A közlemény virágnyelven tudatja velünk, hogy a „számítástechnikai és alkalmazásfejlesztési témák objektív, elemző feltárásának hagyományát" tovább folytatni jelenleg csak elektronikus formában lehet. A Byte élt 22 évet. Életének utolsó periódusában is megjelent 550 ezer példányban. Az olvasók szerették. Ők igen. Ja, hogy nem nekik kellett volna készíteni?!



Paint* és Effect* 2.0

A videografika Discreet bája

A Discreet Logic korábban Illuminare Studióként megvásárolt egy videopaint programot és egy kompozíciós eszközt, amely lehetővé teszi a felhasználók számára, hogy kihasználják a 3D tér előnyeit. Nemrégiben kezdte a cég szállítani a középkategóriás szoftverpáros továbbfejlesztett változatát, most már nemes egyszerűséggel Paint* és Effect* 2.0 néven.

A Paint* egy vektor alapú 2D grafikai (paint) alkalmazás, amelyben egyaránt dolgozhatunk álló- és mozgóképekkel. A 2.0 verzió számos új funkciót tartalmaz Macintosh és PC rendszerre egyaránt. Az egyik legfontosabb fejlesztés az interaktív kapcsolat a 3D Studio Max szoftverrel, az Open API architektúra segítségével. A Max felhasználók ezáltal statikus és animált textúrákat készíthetnek, mindvégig közvetlenül a Maxon belül, a modellen ellenőrizve az eredményt. Fő előnye az ilyen jellegű textúrakészítésnek, hogy vektoros kép lévén nem függ a felbontástól. A Paint* álló- és mozgóképekkel egyaránt dolgozik, és egyszerűen lehet animált textúrát készíteni, amely szabadon kinagyítható, anélkül, hogy veszítene részletgazdagságából.

Keze(lé)s

Nemcsak a Max funkció miatt érdemes a Paint*-et számításba vennünk. Önmagában is rendkívül sokoldalú képmanipuláló eszköz. A munkafolyamat objektumorientált, vagyis minden munkafázis önálló eseményként jelenik meg az időgörbén. A lépések így animálhatóak, a „timeline" segítségével sorrendjük felcserélhető. Az effektek csak az alattuk lévő lépésekre vannak hatással, fel-le mozgatva hatókörük bővíthető vagy szűkíthető.

Az új változat felülete kitisztultabb, a használatot segítő számos funkcióval egészült ki. A funkciópaletták szétoszthatók, kombinálhatók, csoportosíthatók vagy több fül alatt összevonhatók, úgy, hogy a legjobban kézreálljanak. Ezzel a lehetőséggel egyszerűen és rugalmasan tudnak hatékony munkakörnyezetet kialakítani, leginkább azok a felhasználók, akik duálmonitoros konfiguráción használják a szoftvereket. A „timeline" a kezelőfelület központi eleme, ez tartalmazza az objektumok hierarchiáját. Szintén az időgörbén kontrolláljuk objektumaink időbeni változásait, és csoportokat is kialakíthatunk például a komplex ecsetvonások menedzselésének leegyszerűsítésére.

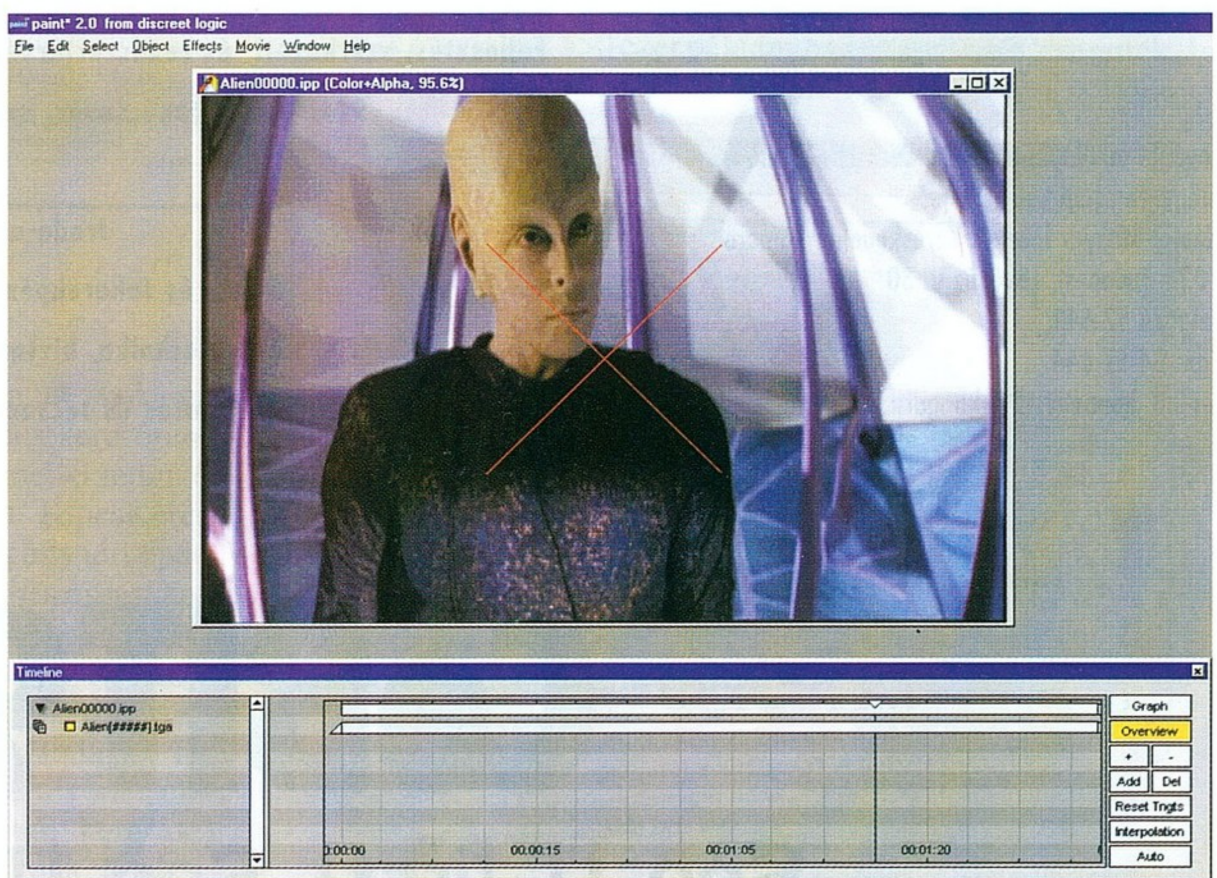
Aki már otthonosabban mozog a grafikai programokban, könnyen boldogul a Paint*-tel is. Néhány extra „kunsztot" azonban tartogat a szoftver. Az egyik ilyen a Colour Mixer, amelyben a szokásos RGB beállítás mellett megtehetjük, hogy valamelyik kiválasztott színhez hozzáadunk egy háttérszínt, majd azt a palettára „rákenve" tovább folytatjuk a színek keverését, mintha igazi palettán és valódi festékkel dolgoznánk.

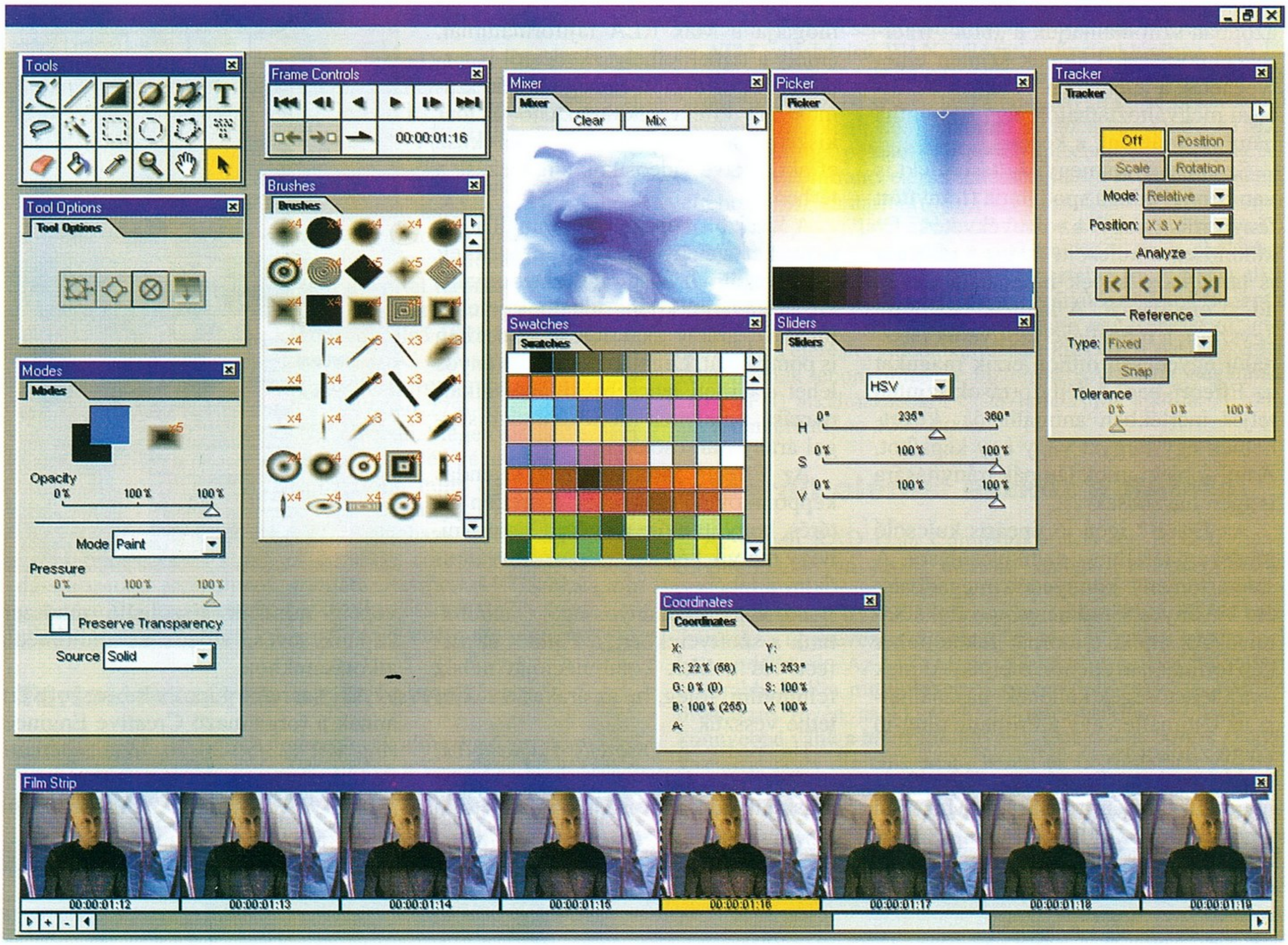
Az ecsettípusokat könnyen elkészíthetjük, megváltoztathatjuk, bár a Paint* rengeteg normál és effekt ecsettel van ellátva. Az ecsetméretet a „Control" billentyű lenyomásával és az egér mozgatásával menet közben is megváltoztathatjuk. Az egyik titkos fegyver a Paint*-ben az átmenet opció. Az átmenet kialakítására és kiszámoltatására a felhasználó teljesen „szabad kezet kap". A szín mellett az átlátszóság és az eloszlás is meghatározható, az animálhatóság (mozgathatóság) pedig igen érdekes effektusokat eredményezhet.

Festőerő

A Paint* igazi erejét a mozgóképek adják. A képszekvencia automatikusan betöltődik, ha az egyik kockára rákattintunk a preview-ablakban, majd egy dialógusablakban kiválaszthatjuk, hogyan akarunk dolgozni a mozival. A megnyitott jelenetet egy filmcsíkon látjuk, míg az adott képkockát a főablakban. Az egyik fontos újítás ebben a verzióban az ún. RAMplayer: a mozit a memóriába lehet kiszámoltatni és onnan optimális sebességgel visszajátszani, ami kiváló előzetes ellenőrzést tesz lehetővé.

A Paint* teljesen animálható, a kulcspozíciók elkészítése és törlése nagyon egyszerű. Ahányszor egy lépést végrehajtunk, egy kulcs kerül az időgörbére. Az átmenetek folyamatosak lehetnek, hiszen „timeline" módból átkapcsolhatunk „graph" módba, ahol Bézier-görbékkel határozhatjuk meg a változások lefolyását. Mivel mind az effektek, mind a rajzelemek objektumok, azok





sorrendje átrendezhető. Így egy nekünk tetsző effekt hatását kiterjeszthetjük olyan rajzelemekre is, melyeket az effekt előtt készítettünk el.

Ne fáradjunk

A Paint* „rotoscope", retusáló eszközei szintén kiterjedtek. A Photoshophoz hasonlóan működik a képtartalom másolása, „klónozása". Lehet igazított, vagy nem igazított, vehetjük a kiindulást ugyanarról a képről, vagy másikról. Amennyiben egy másik képszekvenciából indulunk ki, a forrás lehet fix képkocka, vagy együtt változhat a cél képszekvenciával. Ami a Paint*-et azonban igazán magasabb szintre emeli, az a Discreet Logic híres képpontkövetés vagy „motion tracking" technikája, amelyet annak idején a cég a zászlóshajójának számító Flame high-end trükkszoftverben mutatott be. Egy videó képelemének nyomon követése és ahhoz a grafikai elem pontos hozzáillesztése igen fontos a munka minősége szempontjából. A Paint* ezt a csúcstechnológiát kínálja: a képsíkot kiválasztva, a „tracker" palettát megnyitva kijelölhető egy képtartomány, amelyet a szoftver gyorsan követ, kockáról kockára. A követés a képkontraszt alapján történik, ezért fontos, hogy a képet kontraszt módban is megnézhetjük — így könnyebb a megfelelő részletet kiválasztani.

Effektálunk

Az Effect*-ben a munka azzal kezdődik, hogy meghatározzuk a felbontást, és hogy 2D vagy 3D kompozíciót készítünk. Utána importálhatjuk klipjeinket egy „preview" ablak segítségével. A betöltött klipeket egymáshoz képest térben is elhelyezhetjük. Ennek elvégzéséhez a felhasználói felületen számos gomb van segítségünkre, például amelyikkel leszűkíthetjük a mozgást egy koordinátára. A felhasználói felületről általánosságban elmondható, hogy a Paint*-hez hasonlóan itt is egyszerűsödött és új funkciókkal bővült. Több klippel dolgozva például hasznos, hogy a „timeline", illetve az objektumpaletta segítségével egyaránt válthatunk köztük.

Számos eszköz segít a térbeli elrendezésben és a klipek szerkesztésében, a „layer" funkcióhoz tartozó „hinge" parancs például egy klip szélét automatikusan a legközelebbi szomszédjához illeszti. Egy másik funkció szülő–gyermek kapcsolatot teremt két vagy több objektum között: ha a szülő mozog, gyermekei is vele mozdulnak, míg a gyermekek a szülőtől függetlenül is elmozdulhatnak. Ez a kapcsolat az objektumpalettán is létrehozható. Aki jártas 3D animációs programokban, más hasonlóságokat is talál. Például null-objektumok (kiterjedés nélküli, referenciaobjektumok) hozhatók létre, amelyek láthatatlanok, de csoportok összefogására, szülőként kiválóan alkalmazhatók. A hagyományos képmanipulációs eszközök közül említést érdemel, hogy projektünknek megfelelő egységes képarányt hozhatunk létre, képeinket a „crop" funkcióval méretre vágva. Hasonlóképpen rendelkezésünkre áll a képek négy pontjuknál fogva történő „felszögelése" effektek létrehozásához.

Mivel térben dolgozunk, kameránk segítségével tárgyainkat bármely irányból, térben nézhetjük. Az Effect*-ben csak egy kamera definiálható, de az szabadon mozgatható. A kamera követhet egy nyomvonalat, amely szintén térbeli görbe lehet. A kamera mindig a görbe érintőjének irányába néz. Az Effect* kamerája igen egyszerű, nincs

például defókusz effektre lehetőség. Ezt azonban szimulálhatjuk a „blur” filterrel, elmosással. Lámpák szintén elhelyezhetők a jelenetben. A képek beállításai megváltoztathatók, úgy is, mintha hátulról átlátnánk a képen.

Számos olyan megvilágítási effekt is van benne, mint a spotlámpa (irányított fényforrás) vagy az árnyékvetés. Ez különösen a kulcsolt, átlátszó részeket is tartalmazó képek használata esetén ad jó eredményt. A lámpák beállítása egyszerű, a modellező szoftverek használói egyenesen otthon érzik magukat az Effect*-ben. A fényforrások szintén teljes mértékben animálhatók, követhetnek egy tárgyat vagy egy képsíkot. A null-objektumok lámpák irányítására is igen hasznosak.

Az Effect* igen jó lineáris kulcsoló rendszert tartalmaz színkulcsolásra. A színtartomány könnyedén megadható, és látványos eredményt produkálhatunk. Az alpha csatornát fekete-fehér kép formájában ellenőrizhetjük. Akinek extra lehetőségek kellenek, használhatja az Ultimatte vagy a Primatte plug-in kiegészítőket is.

Akárcsak a Paint*, az Effect* is optimalizálva van a 3D Studio Max-szal való együttműködésre. Az Effect* támogatja a Max RLA fájlformátumát, amely a térbeli elhelyezkedésről is tartalmaz információt. Ennek alapján számos 3D effekt alkalmazható a már kiszámoltatott klipeken: köd, defókusz effekt, illetve videosíkok elhelyezése a térben.

A kirajzolt képeken kombinálhatjuk is ezeket a hagyományos effektekkel: ködre alkalmazva spiráleffektet organikusan kavargó ködöt adhatunk jelenetünkhöz, amely akár furcsa színekben is pompázhat. Legalább ilyen látványos lehet a térbeli lencsebecsillanás alkalmazása, amelynek mértéke a távolsággal arányosan csökken.

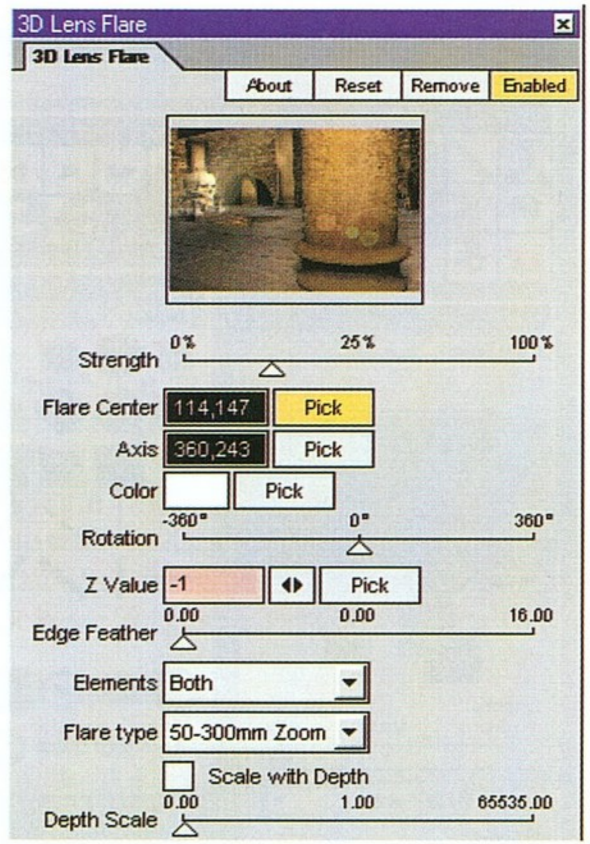
Az Effect* tartalmazza a legendás képpontkövető eljárást is. Egyetlen eltérés, hogy itt előre meg kell határozni, hogy melyik képsíkot fogjuk hozzárendelni a leképezett mozgáshoz.

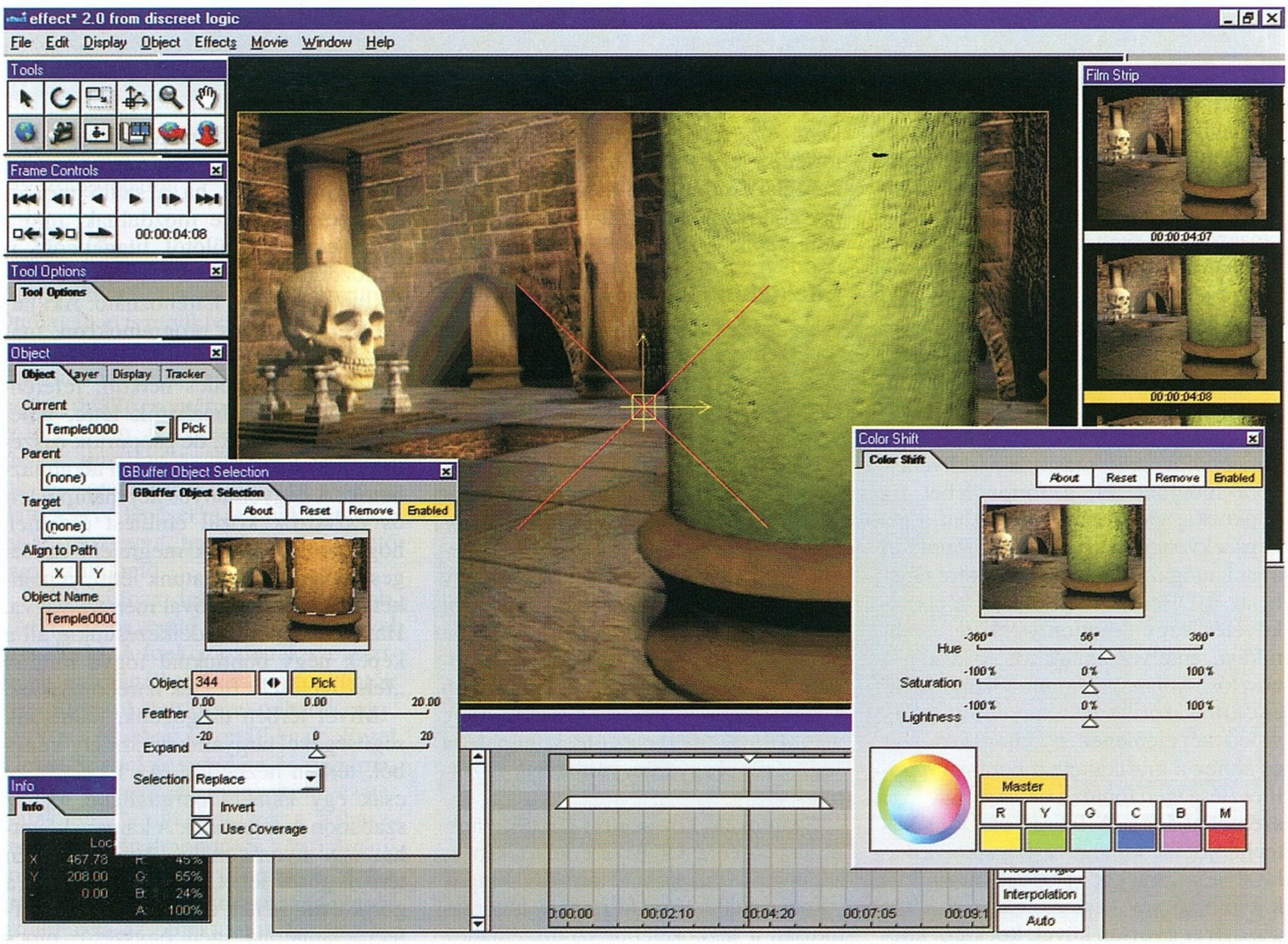
Bár sok apró hiányosságot is találhatunk a szoftverekben, a Paint*, az Effect* és a Max kombinációját nehéz felülmúlni, főleg, ha az árakat is tekintetbe vesszük.

Ha pedig a Discreet Logic NLE szoftvert, az Edit*-et is a rendszerbe illesztjük, akkor a produkció elejétől a végéig mindent kiszolgáló rendszert kapunk tévés, videós és multimédia alkotásainkhoz.

Aki szeretné kipróbálni a szoftvert, annak a forgalmazó Creative Engineering Kft (23/50-50-70, media@creative.hu), ingyenes demó-CD-t ad.

Bátor Csaba





Piacvezető a Nortel Passportja

Az IDC, a Dell'Oro Group, az In-Sat, a Dataquest és a Vertical Systems Group (VSG) külön-külön készített értékelései egyaránt első helyre sorolták a Nortel Networks cég Passport nevű, multiszolgáltatású, ATM és Frame Relay (FR) kapcsolóit. Az IDC szerint a Nortel részesedése a világ WAN ATM és FR kapcsolóinak piacán 19,5%-ot ért el 1997-ben. Ez a piaci szegmens 2002-re 160%-kal növekszik, és eléri a 7,6 milliárd dolláros forgalmat. A Dell'Oro és az In-Sat piaci előrejelző cégek szerint 1998 első felében a multiszolgáltatású kapcsolók piacán 32, illetve 44 százalékkal a Nortel termékei vezették az eladások listáját. A VSG szerint a Passport switch az ATM alapú vállalati WAN kapcsolók között volt első 1998-ban, 34 százalékos részesedéssel. Ugyanakkor a VSG a Passportot 39%-kal az FR vállalati kapcsolók vezetőjeként nevezte meg. A Dataquest cég szerint a Nortel Networks 1997-ben 27%-os részesedéssel világelső volt a FR ATM eszközök piacán. Megerősítendő ezeket a statisztikákat, a Nortel Networks a múlt év végén bejelentette, hogy világszerte eddig összesen több mint 20 ezer darabot értékesített a Passport termékcsaládból, amely jelenleg a következő tagokból áll: Passport 6000: skálázható, multiszolgáltatású vállalati kapcsolók. Passport 7000: ún. core kapcsolók megoldássorozata a szolgáltatók számára. Passport 4400: multiszolgáltatású fiókirodai elérési eszközök. Passport 4120: hajlékony adathálózati platform a LAN optimális adatkonszolidálására.

Tartalomszűrők az Internethez

Az Icon Kft által kínált, Internetre alkalmazható tartalomszűrő megoldások fő célja a vírusok, illetve a veszélyes kódrészeket és kódscripteket hordozó programok elleni védekezés a világhálón. A Trend Micro Inc. cég InterScan VirusWall nevű programja az SMTP, HTTP és FTP forgalom vírusellenőrzésére szolgál, támogatja a vírusminták automatikus frissítését és webböngészővel történő felügyelésre alkalmazható. A támogatott rendszerplatformok: Solaris (Sparc), Solaris (Intel), Windows NT, HP-UX. Alkalmazásával többek között szavak, szóösszetételek keresése, spam-ok kivédése oldható meg a világhálón. A cég másik terméke, az InterScan WebProtect Microsoft proxyszűrőként működik. Legfőbb erénye, hogy az FTP vagy HTTP állományokból képes az esetleges vírusokat felismerni és kiirtani, blokkolja a veszélyesnek és „rosszindulatúnak" tűnő ActiveX objektumokat és Java appleteket, továbbá blokkolja az alá nem írt, vagy ismeretlen hitelesítésű Authenticode programokat. Az Authenticode Manager csomag szintén elsősorban az e-mail, és csoportmunka szoftverek és a webforgalom ellenőrzését végzi a ScanMail programmal. Kifejezetten ajánlják MS-Exchange, Lotus Notes, cc:Mail, MS-Mail és HP Open Mail levelezőrendszerekhez. Az Authentic Manager adatbázisok és replikációs folyamatok valósidejű vírusellenőrzését is képes elvégezni. Az Icon a Content Technologies cég Mimesweeper tartalomszűrő és hálózati (Internet, intranet) erőforrásvédő csomagját is forgalmazza. A csomag két autonóm modulja (Mailsweeper, Websweeper) MS-NT alatt futtatható. A vírusölő programok legtöbbjét a Mimesweeper támogatja, azt sok esetben tűzfalprogramokkal integráltan is forgalmazzák. A Mimesweeper szoftver korlátok között tartja a Web használatát, de csak minimális mértékben lassítja le az internetes munkát.

Cisco routerek fiókirodáknak

1998 december végén kezdte szállítani új, nagysebességű útválasztó berendezéseit a Cisco Systems. A digitális előfizetői vonalakra (DSL) tervezett újgenerációs Cisco 1400-as router családot elsősorban kis és közepes üzleti forgalmat lebonyolító cégeknek, illetve kis fiókirodáknak ajánlják. A sorozat első tagja, a Cisco 1401 DSL, nagysebességű Internet/intranet elérést kínál, biztonsági funkcióinak és szolgáltatásminőségi (QoS) jellemzőinek további javításával. A Cisco 1400-as sorozat a Cisco végponttól végpontig terjedő DSL hálózati megoldásának kulcseleme. Az 1401-es „doboz" egy 10BaseT Ethernet és egy standard ATM25 csatolóporttal rendelkezik. Az ATM25 interfész az ATM cellabázisú WAN hálózati infrastruktúrához való láthatatlan kapcsolódást biztosítja, továbbá a világon a legtöbb postai és távközlési szolgáltató által igényelt külső demarkációs pontként is szolgál; kombinálható továbbá egy külső ATM25-DSL modemmel, például a Cisco626 típusúval, annak érdekében, hogy Internet/intranet elérést lehessen max. 8 Mbit/s sebességgel létrehozni és fenntartani. Az 1401 használata a gyártó szerint kis- és közepes vállalkozásoknak fontos ATM és multiprotokoll előnyöket jelent, beleértve a QoS funkciót és a skálázható sávszélességet. Ezek révén olyan új osztályú multimédia alkalmazások generálhatók, amelyek integrálják az adatot, hangot és a videót. A Cisco az 1401-et leginkább nagysebességű Internet/intranet elérésre, videoalkalmazásokra (VoD), desktop videokonferenciák lebonyolítására, biztonságos virtuális magánhálózati (VPN) alkalmazásokra és távtanuláshoz ajánlja. A router ára a tengerentúlon 1395 dollár.

Integrált elérési eszköz a RAD-tól

Az izraeli RAD Data Communications cég FCD-IP hozott forgalomba olyan távoli elérésre szolgáló integrált hálózati eszközt (IAD), amely adat- és faxátvitelhez PBX csatolóegységgel, beépített IP/IPX alapú routing funkciókkal, E1/T1 bérelt vonalakon úgynevezett „in-band" és távfelügyeleti képességekkel rendelkezik. Az IAD céljára kialakított egység helyettesíti a korábbi külön multiplexer és router dobozokat. A beépített útválasztó révén csökkennek az infrastruktúra költségei, és könnyebbé válik a hálózatfelügyelet. Opcionális portok szolgálnak arra, hogy digitális adat-, hang- és faxforgalmat az E1/T1 szolgáltatásba lehessen állítani, ezáltal az E1 „csővonalak" minden előnyét ki lehessen használni. Az FCD-IP egység tartalmaz továbbá két LAN portot is az IP/IPX jellegű forgalom útvonalválasztására. A berendezés további jellemzői közül megemlíthető, hogy az FCD-IP saját routere rugalmasan használható a hálózatban meglévő routerekkel, és a szolgáltatók figyelhetik a terhelést, az E1/T1 csatorna jobb kihasználása érdekében. Egy automatikus mechanizmus (sub E1/T1 bypass) megbízható, hibamentes hangátvitelt biztosít. Az új termék kiegészíti a RAD eddigi széleskörű FCD interfész-választékát. Minden egyes FCD egységet ugyanaz a menedzselő programrendszer vezérli.

SCO: Tarantella és Windows szerver

Az SCO cég Tarantella programcsomagja a hálózat alapú számítástechnika első alkalmazásügynöke (application broker), amely lehetővé teszi a felhasználóknak, hogy bármilyen Java-futtató böngészővel vagy Windows PC-vel tetszőleges Unix, IBM 3270 vagy Windows alkalmazást elérjenek, újabb ügyféloldali szoftver telepítése nélkül. A Tarantella így lehetővé teszi az alkalmazások Weben keresztüli azonnali elérhetőségét, anélkül, hogy az alkalmazásokat újra kellene írni. A Tarantella új, 1.3-as verziója nagyon megkönnyíti a rendszergazdák munkáját azáltal, hogy a windowsos kliensek számára sokkal egyszerűbbé teszi a Windows terminálszerveren (WTS) futó alkalmazásokhoz való hozzáférést. Ezenkívül nagyobb biztonságot és gyors adatelérést tesz lehetővé. A felhasználók alkalmazásaikat a Windows desktop rendszerekről WTS szerverekre helyezhetik át, hogy elkerüljék az asztali Windows rendszereken futó, erőforrásigényes ügynökprogramok által okozott problémákat. A Tarantella 1.3 révén a rendszergazdák a számos kiszolgálón futó alkalmazásokat központi helyről tehetik elérhetővé és felügyelhetővé egyszerű drag-and-drop művelettel, hasonlóan ahhoz, ahogy ma a Tarantella a unixos és nagygépes alkalmazásokat kezeli. Ez a megoldás jelentős időnyereséget hoz a komplex windowsos alkalmazáskezeléshez képest. A Tarantella 1.3 már számos platformra kapható, beleértve a HP-UX, IBM AIX, Sun Sparc Solaris, UnixWare 7 és SCO UnixWare rendszereket. A Tarantella listaára felhasználónként 395 dollár.

Kovács Attila

NetWare for Small Business 4.2

Kisvállalatok nagy lehetősége

A Novell 1998 decemberében kezdte meg a kisvállalati hálózatokra tervezett NetWare for Small Business új, 4.2-es verziójának forgalmazását. Ennek belső magja, kernelje azonos a közel egyidejűleg piacra került NetWare 4.2 operációs rendszerével. Mindkettőnek alapja a NetWare 4.x rendszer, viszont ezek a változatok már biztosítják a 2000. évi dátumváltás kapcsán keletkező problémák kiküszöbölését, illetve tartalmazzák az eddigi javításokat, fejlesztéseket.

Az új kisvállalati NetWare verzió általában olyan hálózatok kiszolgálására alkalmas, amelyekben egy szerverre 5-20 munkaállomás csatlakozik. Az egyidejűleg hozzákapcsolható felhasználók maximális számát a korábbi 25-ről 50-re növelték. A telepítéshez a szerver oldalán legalább 100 MHz-es Pentium és 48 MB RAM szükséges. A merevlemezterület — más szervergépekhez hasonlóan — az elvégzendő feladatoktól függ.

Alkalmazások, segédeszközök

A 4.2-es szoftvercsomag egyúttal nyitási lehetőség a NetWare 5-ös alapú rendszerek felé is. Megtalálható benne például egy SQL alapú, ötfelhasználós Oracle 8 adatszerver, amely hatékony eszköz a kisebb vállalkozások adatbáziskezelési feladataihoz, vagy a Z.E.N. Starter Pack, amely azonos a Z.E.N.works-nek a NetWare 5-ben is meglévő kezdőcsomagjával. Ez utóbbi használata szorosan kötődik a Novell címtári rendszerhez (NDS, Novell Directory Service), amely biztosítja a hálózati erőforrások könnyű, központosított menedzselhetőségét. Ez azért is fontos, mert a kisebb hálózatok üzemeltetése nem nagyon „bírja el" külön informatikai csapat alkalmazását.

A rendszer használatát többféle program segíti. Rögtön a telepítésnél rendelkezésre áll a QuickStart, amely nagyfokú automatizmussal képes levezényelni az új rendszer teljes installálását. A hálózat munkaállomásainak adminisztrációjában, menedzselésében legfőbb segítség a már említett Z.E.N.works, és van hozzá még egy fontos segédlet, a Novell Easy Administration Tool (NEAT). A mindennapi munka üzenetváltásaihoz az egyik segédeszköz a Tobit FaxWare 5, faxüzenetek küldésére és fogadására közvetlenül a jogosult munkaállomásokról; a másik a GroupWise rendszer a dokumentumok és a naptárbejegyzések kezelésére, a belső levelezésre, az elektronikus üzenetek megosztására — függetlenül attól, hogy azok szövegszerkesztővel készült vagy e-mail dokumentumok. (Ez utóbbi az Internet technológiáját hozza be a kisvállalkozások belső hálózatába is.)

Csatlakozás az Internetre

Az Internetre való csatlakozást a NetWare for Small Business 4.2 többféle szolgáltatása, segédprogramja segíti. A világhálóval a kapcsolat kialakítását megkönnyíti a termék beépített varázslója (Internet Connection Wizard), de ezen kívül is többféle — részben vagy teljes egészében más-más fejlesztőműhelyekből származó — program áll rendelkezésre. Ezek egyike a korábbi verzióhoz (4.11) is mellékelt Netscape FastTrack Server for NetWare. Az Intel alapú rendszereknek ez az egyik leggyorsabb webszervere, amely az új verzióban is biztosítja a helyi hálózatban készített dokumentumok gyors publikálását a Neten.

A dokumentumok fogadási oldalán a hálózati operációs rendszer Netscape Navigatort ad böngészőprogramként. A vállalkozás internetes kapcsolattartásának gördülékeny lebonyolítását elősegítő többi program közül a Novell BorderManager FastCache biztosítja a webszerver elérésének és a böngészésnek a gyorsítását. Ezzel javulhat azon munkák hatékonysága, melyekben az Internet lényeges szerepet játszik.

Bizonyos esetekben a válaszidő a külső adatkommunikációban különösen fontossá válik, például amikor egy vállalkozás bekapcsolódik az elektronikus kereskedelembe, akár olyan alapszinten is, hogy saját üzleti ajánlatát, katalógusát önálló weblapon szerepelteti. A saját honlap elkészítéséhez és „üzemeltetéséhez" hasznos segédeszköz a NetObjects Fusion 3.0, mindenek előtt a weblap megtervezésében és kivitelezésében, de később is, amikor rendszeresen kell frissíteni a cég weboldalait.

Adatbiztonság

A rendszer adatvédelméhez szintén vannak hatásos eszközök az új hálózati programcsomagban. A hálózati mentések a grafikus felhasználói felületen futó Sbackup segítségével oldhatók meg, akár a szerver, akár a kliensgép adatállományairól van szó. A rendszer vírusvédelmét a korábbi verzióhoz hasonlóan külső szoftverek biztosítják. Ezek azonban most a Network Associates NetShield és VirusScan termékei, melyek licencét automatikusan megkapja az, aki a kisvállalati NetWare csomagot megvásárolja.

Ha a hálózatot használó kisvállalkozásból nagyobb lesz, vagyis kinövi a NetWare for Small Business 4.2 által biztosított kereteket, rugalmas lehetőség van a hálózat további bővítésére. Bár az új NetWare 5 operációs rendszer kernelén alapuló kisvállalati rendszerek későbbi megjelenésének ütemterve jelenleg még nem ismeretes, a Novell biztosítja az áttérést a NetWare 5-ösön alapú hálózatra is.

Simay Endre István

NetWare for Small Business 4.2 Server Plus 5-user Connection License (German)	00662644246703	A	$1,295.00
NetWare for Small Business 4.2 Server Plus 5-user Connection License (Spanish)	00662644246710	A	$1,295.00
NetWare for Small Business 4.2 1-user Connection License (EPFIGS)	00662644246727	A	$70.00
NetWare for Small Business 4.2 5-user Connection License (EPFIGS)	00662644246734	A	$350.00

Tivoli, az univerzális menedzser

Mindenkivel szót értve

Csaknem pontosan három évvel ezelőtt döntött úgy az IBM, hogy megszerzi a Tivoli százszázalékos tulajdonjogát. Elegendő idő telt el tehát ahhoz, hogy értékelhető legyen: mik voltak ennek a tulajdonosváltásnak a pozitív és negatív következményei. Az alábbiak erre adnak választ — „jól értesült körökből".



Minden döntésnek vannak előnyös és hátrányos következményei, de a Tivoli megvásárlása igazi sikertörténet lett a lassan mindennaposnak számító cégösszeolvadások sorában. Kifejezetten jóindulatú felvásárlásról volt szó, hiszen a Tivoli maga is aktívan keresett komoly, tőkeerős befektetőt, hogy megtarthassa versenyképességét. Az IBM révén jelentős anyagi, kereskedelmi és fejlesztési erőforrásra támaszkodhattak, és ennek meg is lett az eredménye. Három évvel ezelőtt a Tivolinak 300 alkalmazottja volt, és 15 millió dolláros évi forgalma. Jelenleg bevételeik elérik a 1,5 milliárd dollárt, és közel 4000 alkalmazottat foglalkoztatnak.

A Tivoli nagy rendszergyártó partnerei — akiknek az IBM az egyik legnagyobb riválisa — eleinte szkeptikusak voltak, arra számítottak, hogy a Tivoli a jövőben az IBM platformokat preferálja majd, és ezekre koncentrálja a fejlesztési kapacitásokat. Valójában ennek az ellenkezője történt: a Tivoli nemcsak szervezetileg maradt független cég, hanem szakmailag is teljes önállóságot kapott az IBM-től.

AZ IBM jól látta, hogy ebben az üzletágban kulcskérdés a platformsemlegesség, és valahányszor új modult fejlesztettek ki, ügyeltek arra, hogy az minden platformon egyidejűleg jelenjen meg. Az első adatbázis-menedzsment termék például ugyanazon a napon lett bejelentve Oracle, Informix, Sybase, Microsoft SQL és IBM DB2 adatbáziskezelőre. Az üzenetkezelő és e-mail menedzsment szoftver is egyszerre jött ki a Lotus Notes és az MS Exchange verzióhoz. A Tivoli rendszermenedzsment modulok új változatai egyszerre jelennek meg Sun, HP, Windows NT és NetWare operációsrendszer-környezetre. Ezzel a következetes fejlesztési koncepcióval sikerült eloszlatni a partnerek aggályait: a Tivoli fő értéke, hogy a felhasználók bármilyen platformon bármilyen alkalmazást tudjanak menedzselni.

Desktop Management Task Force

A platformsemlegesség kérdése szorosan összefügg a nyíltsággal, a hivatalos és az ipari szabványok támogatásával. A rendszermenedzsment területén a Tivoli is felsorakozott bizonyos szabványosítási törekvések mellé.

Az iparág legjelentősebb szabványa az ún. Common Information Model (CIM) specifikáció, amely valamennyi platformon közös struktúrát definiál a menedzsment információk tárolására. Ezt a szabványt a Desktop Management Task Force (DMTF) nevű szervezet fejleszti, amelyben a Tivoli aktívan tevékenykedik az Intel, a Microsoft, a Sun, az IBM képviselőivel együtt. A CIM jó úton halad afelé, hogy hivatalos általános szabvánnyá váljon, jelenleg a szabványtervezet kettes változatán dolgoznak, és várhatóan 1999-ben el is fogadják azt. Amint meglesz a végleges verzió, beépül a Tivoliba, és többek között a Microsoft is támogatni fogja a Windows NT 5-ben.

Tendenciák a fejlesztésben

A Java szintén szerepel a cég fejlesztési elképzeléseiben. Ma is használják a Javát a Tivoliban, mind a kliens-, mind a szerveroldalon. Több termék felhasználói felülete készült Javában, hogy webes környezetben is futtatható legyen. Egyebek mellett a már piacon lévő Tivoli Global Enterprise Manager (GEM) alkalmazás teljes egészében Javában íródott. Szerveroldalon kiemelendő a kisebb vállalkozások számára készült Tivoli IT Director, amely 80 százalékban Javában, 20 százalékban C++-ban van implementálva.

A webes környezetben futó menedzsment konzolokról szólva az a kérdés, hogy ezek végképp felválthatják-e a dedikált menedzser-munkaállomásokon futó hagyományos Windows vagy Unix alapú klienseket. A Tivoli is támogatja a Microsoft által szorgalmazott WEBGM szabványt — ez elvileg a vállalat bármely olyan munkaállomásáról hozzáférhetővé tesz bizonyos diagnosztikai és felügyeleti funkciókat, amelynek van Java-képes böngészője. Ugyanakkor azt is tudni kell, hogy a webes és a kliens-szerver alapú menedzsment konzolok között ma még igen nagy a különbség az adatbiztonság tekintetében. Semmi akadálya viszont, hogy a helyi rendszergazdák vagy maguk a felhasználók egy böngészőn keresztül analizálhassák a rendszer egyes erőforrásainak terhelését, lekérdezhessék a rendszerparaméterek állapotát, vagy kisebb konfigurációs beavatkozásokat végezhessenek. A rendszer legérzékenyebb pontjaiba azonban még jó ideig csak a hagyományos kliens-szerver alapú, dedikált központi menedzsment munkaállomásokon futó konzolokról lehet beavatkozni.

Tivoli Management Agent

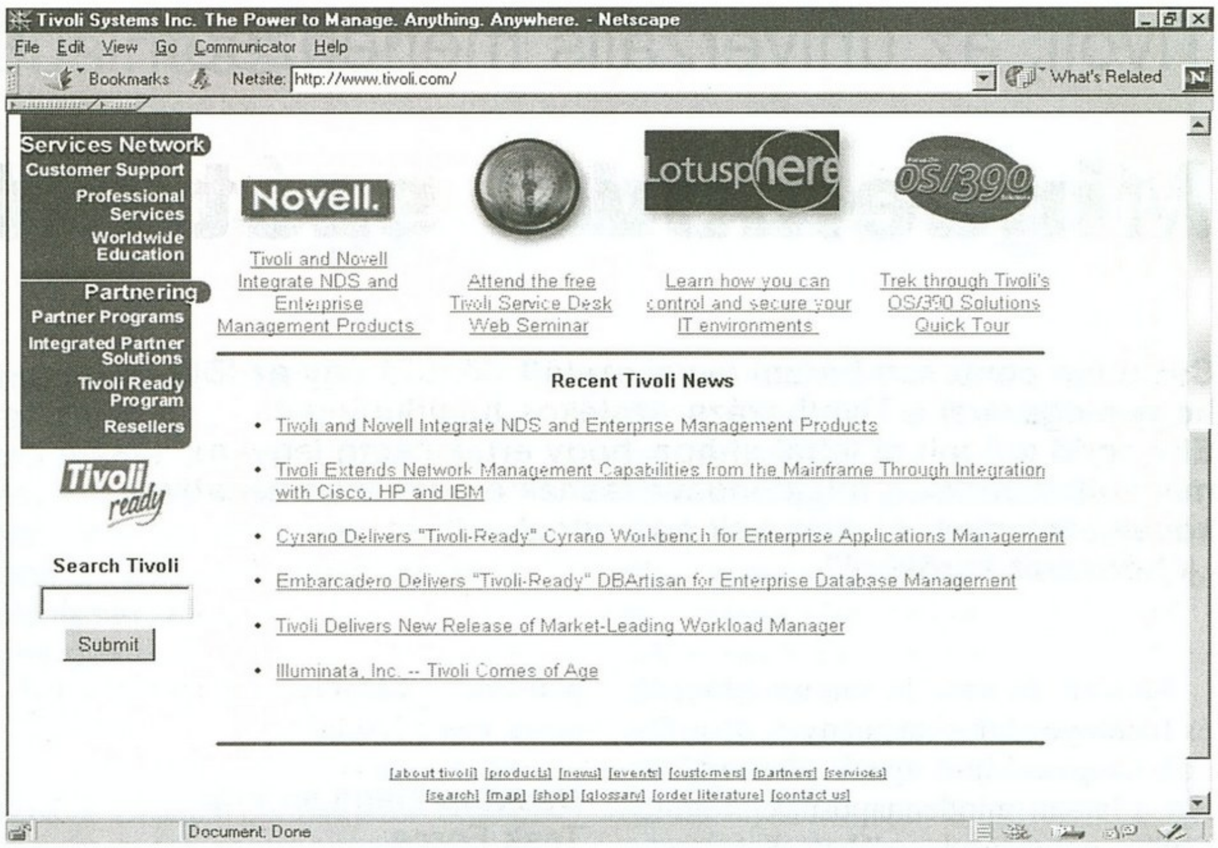
A menedzsment szoftverek első generációja még csak a hálózati eszközök távoli felügyeletére koncentrált, és az SNMP ügynökmodulokon keresztül érte el a hálózat távoli komponenseit. Ez a koncepció centralizáltságot jelent, a végpontokra nem telepít intelligenciát, csak a szabványos SNMP ügynökmodult. E technológia előnye, hogy az SNMP széles körben elfogadott szab-

vány, az SNMP ügynökmodul gyakorlatilag minden operációs rendszerben és hálózati eszközben működőképes, így az eszközök már az üzembe helyezés pillanatában menedzselhetővé, távolról elérhetővé válnak. Ugyanakkor hátránya, hogy korlátozott az SNMP által támogatott funkcionalitás, nincs benne felhasználói szintű menedzselési lehetőség, nem tartalmaz megfelelő biztonsági szolgáltatásokat. Mindezek miatt elsősorban hálózati eszközök menedzselésére és monitorozására alkalmas, az operációs rendszer és az alkalmazások felügyeletére csak igen kis mértékben.

Másik megközelítési mód szerint a vállalati szintű rendszermenedzsmentet önmagában is elosztott, többrétegű hálózati alkalmazásnak tekintjük, és a végpontokon bonyolult kliensmodulokat futtatunk. Ezekbe minden szükséges funkcionalitás beépíthető, ami a magas szintű szolgáltatásokhoz szükséges. Kétségtelen azonban, hogy egy több ezer vagy több tízezer munkaállomással fölszerelt nagyvállalat esetében a rendszermenedzsment környezetnek mint komplex alkalmazásnak a telepítése, futtatása, karbantartása önmagában is nagy feladat, és nem automatizálható.

A Tivoli új verziójában a megjelenő dinamikus ügynöktechnológia jelentősége abban rejlik, hogy egyesíti a fent vázolt két megközelítési mód előnyeit: az SNMP egyszerűségét és a klasszikus kliens-szerver menedzsment architektúra gazdag funkcionalitását, biztonságát, rugalmasságát. Kifejlesztettek egy univerzális, kis ügynökmodult, a Tivoli Management Agentet, ami az egész rendszer lelke, és ezt kell telepíteni az összes IT eszközbe, amelyet be szeretnénk kapcsolni a menedzsment rendszerbe. Egyedül ezt az apró modult kell telepíteni, és csak egyszer. Az összes többi kliensmodul automatikusan letöltődik a végpontokra, valahányszor a szerveroldalon új szolgáltatást helyeztek üzembe, vagy frissítettek egy meglévőt.

Hangsúlyozandó, hogy a kliensek letöltődése dinamikusan, az igénybe vett menedzsment szolgáltatások függvényében történik, és teljesen automatikusan, vagyis a rendszergazda csak a szerveren installálja az újabb szoftverkomponenseket, a kliensekre minden szükséges modul automatikusan települ. Körülbelül olyan könnyebbséget jelent ez a technológia a menedzsment infrastruktúra működtetésében, mint amit az egész rendszer nyújt a magas szintű alkalmazások vállalati szintű telepítésében, üzemeltetésében.



És még valami: a Tivoli Management Agent gyorsan és könnyen portolható a különböző platformokra, nem foglal el sok helyet, nem köt le jelentős erőforrásokat, ezért életképes törekvés, hogy minden elterjedt operációs rendszer és hálózati eszköz standard részévé váljon. Így — akárcsak az SNMP esetében — abban a pillanatban, hogy egy új eszközt illesztenek a vállalati hálózatba, az automatikusan a Tivoli menedzsment környezet eleme lesz. Tekintve, hogy a számítástechnikai eszközök néhány év alatt jelentős mértékben kicserélődnek, minden esély megvan rá, hogy a világ nagyvállalati számítástechnikai eszközparkja rövid idő alatt „Tivoli-kész” legyen.

Integrációs trendek

A Tivoli az IBM jóvoltából kiegészíthette palettáját a Netview hálózatmenedzsment eszközzel is. A Netview integrálásának első szakaszában esemény szintű integráció valósul meg, hogy a Netview hálózatmenedzsment és a Tivoli rendszermenedzsment moduljai között lehessen eseményeket továbbítani.

A második fázis az adatok, objektumok és eljárásmódok szintjén történő integráció, hogy a Netview és a Tivoli alkalmazások ugyanazokat az adatszerkezeteket használják, így hozzáférhessenek egymás adatbázisaihoz és objektumaihoz.

A harmadik szint a grafikus felhasználói felületek teljes integrációja, az első két integrációs lépés sikere után a fejlesztők jelenleg már ezen dolgoznak.

Egyes elemzők szerint a rendszermenedzsment termékek egyre inkább integrálódnak az operációs rendszerekbe, és akár el is tűnhetnek mint önálló alkalmazások. Az mindenesetre tény, hogy egyre több alapvető monitorozási és konfigurációs funkció beépül az operációs rendszerbe. A fejlesztés így másra koncentrálhat: az alkalmazások, a különböző platformok között megvalósítandó magasabb szintű együttműködés funkcióira. De nem valószínű például, hogy a Sun valaha is beépít a Solarisba olyan technológiákat, amelyekkel könnyebb lesz az NT szerverek menedzselése. Az Oracle sem fog beépíteni olyan funkciókat az Oracle RDBMS-be, melyek a Microsoft SQL szerver adminisztrációját teszik hatékonyabbá. A nagyvállalati felhasználóknak viszont általában párhuzamosan kell alkalmazniuk különböző gyártók operációs rendszereit és alkalmazásait, ezért mindig is szükségük lesz olyan termékekre, amelyek képesek ennek a heterogén környezetnek az integrált menedzselésére.

A következő fontos lépés, hogy az integrált menedzsment környezetekbe beépül a szolgáltatási szintek menedzselésének támogatása (Service Level Management). Ez üzletközpontú szemléletmódot jelent, amikor a rendszermenedzsment szoftvernek már nemcsak az a dolga, hogy a vállalat IT részlegének a munkáját megkönnyítse, és ezáltal csökkentse az üzemeltetési költségeket, hanem ellenőrizhetővé és mérhetővé kell tennie a vállalati IT szolgáltatások színvonalát, valamint a megfelelő szolgáltatási szint eléréséhez szükséges ráfordításokat. Ebben kulcsszerephez fognak jutni az ún. help-desk rendszerek, amelyek az integrált menedzsment környezetek szerves részévé válnak.

Hutter Ottó

Microsoft kontra Linux

A szabad forráskód csatája

Egy 1998. augusztus 11-i dátumú Microsoft feljegyzés, az ún. Halloween Document részletesen tárgyalja az OSS (Open Source Software, a szabad forráskódú szoftverek) és a Microsoft viszonyát. Egészen pontosan azt, hogy milyen veszélyt jelentenek a szabadszoftverek a világ legnagyobb szoftvergyártójára — és persze azt is, hogy mit lehetne ellenük tenni. Feltehető persze a kérdés, hogy egyáltalán mi bizonyítja a dokumentum létét és hitelességét. Bizonyítani mi sem tudjuk, de az tény, hogy az utóbbi időben nem egy szakértő valódinak tekintette. Az is figyelemre méltó, hogy nem kisebb szaktekintély, mint Eric S. Raymond, az autentikus forrásnak számító (és az említésre méltó hacker bölcsességeket összefoglaló) Jargon File szerzője is reagált a dokumentumra, kár lenne hát szó nélkül elmennünk mellette.

Nem is olyan régen még nyugodtan „lesajnálhattuk" azokat az Interneten rendszeresen olvasható élcelődéseket, hogy Bill Gates álmatlan éjszakáit a Linux okozza. A Linuxot (különösen pedig a kizárólag Linuxot) használók száma ma is legfeljebb a töredéke azokénak, akiknek gépén az MS Windows valamelyik változata fut.

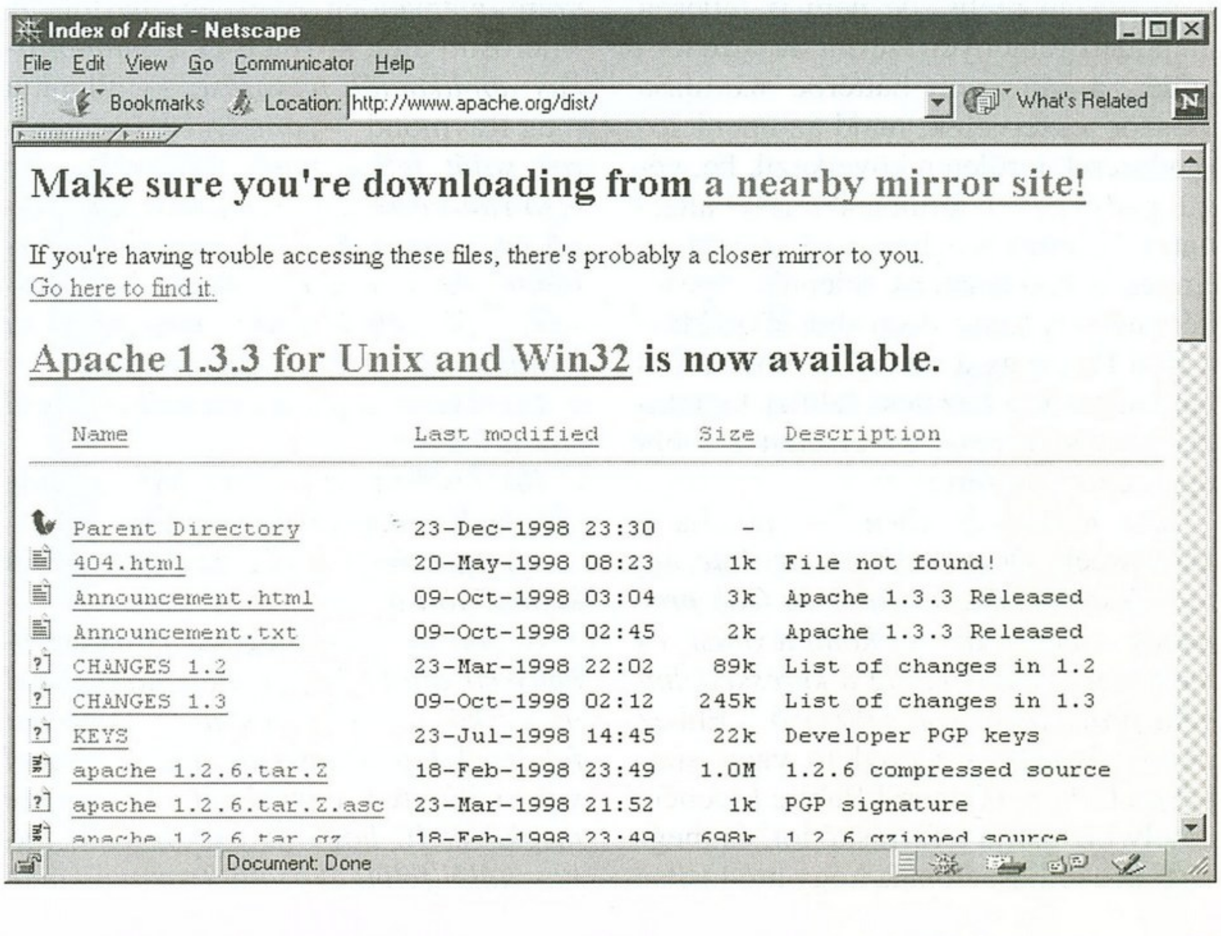
Élen az Apache

Nem biztos azonban, hogy az OSS jelenség ma is ugyanaz, mint néhány éve volt. A webszerverek világában ugyanis a szabadszoftverként fejlesztett Apache 44,6%-os részaránnyal áll az első helyen, míg a Microsoft szervere 29,4%-os (igaz, növekvő tendenciájú) részesedéssel a második helyen. A Netscape csupán 9,5%-ot tudott magának kihasítani a tortából, az O'Reilly pedig alig 4%-ot. Az NCSA a maga 1,1%-ával szinte már említésre sem lenne érdemes, illetve ha mégis, akkor leginkább azért, mert 1995 februárjában még az általa kifejlesztett szerverszoftver, a public domain HTTP Daemon volt a listavezető, csakhogy 1994 közepe óta egyetlen bájtnyi kódot sem írtak át, és egyetlen hibát sem javítottak ki benne, ezért az elégedetlen rendszergazdák elhatározták, hogy saját kezükbe veszik az ügyet. Nyolcan megalapították az Apache Group-ot, és az 1995 április közepén kibocsátott Apache 0.6.2 rögtön be is futott (www.apache.org).

Hogy árnyaltabb legyen a kép, hozzá kell tennünk, hogy a Ziff-Davis felmérése szerint az Interneten szolgáltatók 65,5%-a legfeljebb 1-4 szervergépet tart fenn, és alig 3,2% az, amely 25-nél többet. Ez azért érdekes, mert éppen a kis ISP (Internet service provider) cégek azok, amelyek az NT helyett a unixos köryezetben működő Apache-ra voksoltak, és mivel ők vannak túlnyomó többségben (a szolgáltatók közel kétharmadának kevesebb mint öt webszervere van), jelenleg lényegében az ő döntésüktől függ, hogy melyik operációs rendszer melyik terméke jut meghatározó szerephez. És valószínű, hogy a Microsoft — legalább is ezen a téren és egyelőre — elveszítette a versenyt.

Kissé általánosabban is megfogalmazva azt mondhatjuk, hogy a köztudottan zárt licencpolitikát folytató Microsoftnak általában is komoly gondot okozhat a szabadszoftver. Ázsiában például azokon a területeken is rohamosan terjed, ahol az Internet előretörése lassabb, és ráadásul általában is egyre többen látnak benne komoly fantáziát. A Caldera vagy a Red Hat azért foglalkoztat programozókat teljes munkaidőben, hogy azok szabad forráskódú (és ennek megfelelően mindenki számára hozzáférhető és módosítható) szoftvereket fejlesszenek. Ez utóbbi vállalkozásban egyébként az Intel 1998 végén kisebbségi részesedést is vásárolt. Megemlíthetjük Amerika hosszú időn keresztül legnépszerűbb szoftverjét is, a jeleleg Corel tulajdonban lévő WordPerfect szövegszerkesztőt, amelynek legújabb, 8-as verzióját Linux alá is kidolgozták, és azt 1998 december 17-e óta szabadon letölthetővé tették. A szerverek nem győzik a letöltést...

A kívülállók egyik legfőbb kifogása a Linux ellen az szokott lenni, hogy bármilyen jó is egy operációs rendszer, ha nincsenek hozzá megfelelő alkalmazási szoftverek — szövegszerkesztők, grafikai programok, áramkörtervezők, lövöldözős játékok stb. —, akkor mitsem ér az egész. Ez a vélemény részben tájékozatlanságot, részben a dolgok valódi értékétől való eltávolodást tükröz. A StarOffice például nagyon sok szak-



ember véleménye szerint mindig is jobb volt, mint a Microsoft Office. A nem sztárolt kiváló szoftverekre azonban többnyire csak akkor figyelnek fel, amikor azokat a mammutcégek felvásárolják — jó esetben azért, hogy hasznosítsák, rossz esetben eleve azért, hogy megfojtsák —, vagy amikor azok komoly befektetéseket kezdenek el vonzani. A nagy szoftverfejlesztők bevonulása mindig lényeges jelzés, és a Linuxszal kapcsolatban most ez figyelhető meg.

Copyleft

A „not supported" sajátos megnyilvánulása az az egyébként érthetetlen, de Magyarországon is sajnálatosan elterjedt hivatali gyakorlat, hogy például a pályázati anyagokat többnyire a legújabb Microsoft Word verzióban teszik fel a hálózatra. Vajon miért? A titkárnők nem ismerik a „Save As..."-t? Így viszont akinek nincs birtokában ez a szövegszerkesztő, azt egyszerűen kizárják a hozzáférésből. A helyzetet egyenesen tragikomikussá tette a Word öninkompatibilitása, hogy az Office 97-es Word formátumban lementett (Word 8-as) dokumentumot nem lehet Word 6-osban megnyitni.

A Microsoft stratégiája a helyzeti előny teljes kihasználásán alapult. Azon, hogy egy olyan világban, ahol a túlnyomó többség úgy is az általa gyártott szövegszerkesztőt használja, a kisebbségnek sincs más választása. A totális monopóliumra alapozott üzletpolitikát a már említett dokumentum szerint a Microsoft most veszélyben látja, különösen, ha még a játékszabályok is megváltoznak. A Linux-pártiak nem igazán reális, de nem is teljesen alaptalan reményei szerint az átrendeződés, a Microsoft háttérbe szorulása először a szerverek, majd az operációs rendszerek területén következik be, végül pedig egyéb területeken is az alternatívák jutnak meghatározó szerephez. Ennek a folyamatnak jelentős lépéseként néhány Linux disztribúció (például a Red Hat vagy a S.u.S.E.) most már a barátságosabb kezelési felület kialakításában is felveszi a versenyt a többi operációs redszerrel.

„Az utóbbi években — mondja a Halloween Document — *az Internet fejlődésének köszönhetően az OSS projektek elérték azt a komplexitást és mélységet, amely eddig a kereskedelmi célú projektekre volt jellemző."* Ehhez persze kellett a Copyleft vagy más néven GPL is (General Public Licence, amely biztosítja a forráskódot, és megengedi a felhaszálónak, hogy módosítsa azt). A hálózaton keresztül egymással kapcsolatot tartó, leginkább a programozás öröméért és az esetleges dicsőségért dolgozó szoftverírók ennek köszönhetően képesek voltak a végeredmény komplexitását tekintve (és programsorokban is mérve) óriási feladatokat megoldani:

Program	Kódsorok száma
Linux kernel (csak x86)	500 000
Apache webszerver	80 000
SendMail	57 000
Xfree86 X Window szerver	1,5 millió
Teljes Linux disztribúció	kb. 10 millió

A statisztikák szerint egyébként az OSS SendMail az összes internetes mailszervereknek kb: 80%-át teszi ki. Mindez ellen a Microsoftnak — már a birodalmi helyzet öntörvényű logikájából következően is — fel kell lépnie. *„Ahhoz, hogy sikeresen küzdhessünk az OSS ellen* — olvasható a Halloween Document-ben — *inkább magát a folyamatot kell támadnunk az egyes cégek helyett."* Amihez Eric S. Raymond hozzáfűzi: *„Ez bizony tényleg fontos felismerés... Az igazi háborúság nem az NT versus Linux vagy Microsoft versus Red Hat/Caldera/S.u.S.E. fronton folyik — a kérdés a szabad vagy a nem szabad forráskód."* Szerinte leginkább arra lehet számítani, hogy a Microsoft kétségtelenül hatékonyan működő propagandagépezete a jövőben nem is annyira egyes „riválisai" ellen irányul majd, hanem inkább az egész OSS jelenséget próbálja meg lejáratni és felmorzsolni.

Kiszorítósdi

A Microsoft termékeinek és a tipikus szabadszoftvereknek a fejlesztési módszere alapvetően eltér egymástól. A Microsoft által követett út a *„monolitikus applikációk tervezése* — állapítja meg Raymond —, *amelyeket meghatároz saját felhasználói felületük... Az első ránézésre felhasználóbarátnak tűnő programok hosszú távon időpazarlónak és energianyelőnek bizonyulnak... fő céljuk pedig meggyőzni a felhasználót arról, hogy a hibák*

(a) részben a szoftverek szükségszerű sajátosságai;

(b) részben az ostoba felhasználók viselkedésének kövekezményei;

(c) egyébként pedig az összes hibát kiküszöböli a következő frissítés...

A másik lehetőség a Unix/Internet/Web által követett megoldás, ahol az eszköz (engine) és a felhasználói felület (UI, user interface) szigorúan el van választva egymástól. Ez persze megköveteli, hogy az eszköz és a felhasználói felület közötti kommunikáció jól meghatározott protokollon keresztül történjen. Kiváló példa erre a megoldásra a böngésző/kiszolgáló (browser/server) viszony."

Ha a Microsoft fel akarja venni a küzdelmet a szabadszoftver ellen, akkor leginkább arra lesz lehetősége (és saját beidegződései is abba az irányba terelik), hogy külsőleg még meggyőzőbb, még monolitikusabb, de egyre inkább az előrehaladott „windowsitis" tünetegyüttesben szenvedő programokat állítson elő. Vagy pedig arra, hogy megpróbálja kisajátítani az eszközök és a felhasználói felület összekapcsolását szolgáló protokollokat, amilyen például a HTTP. (Erről mindenkinek a Java juthat az eszébe.) A Halloween Document szerint *„a Linux csak addig lehet győztes, amíg a szolgáltatások és protokollok mindenki számára hozzáférhetőek"* és *„ezen protokollok továbbfejlesztésével, illetve újak létrehozásával ki lehetne szorítani az OSS projekteket... Felemeljük a lécet és megváltoztatjuk a játékszabályokat"* — fogalmazza meg a dokumentum szerzője. Magyarán, ha lesöpörhető a pályáról a TCP/IP, az SMTP, a HTTP, a POP3 és az összes többi „open standard", akkor vége a játszmának, és megint a Microsoft nyer.

Legalább is a Microsoft szerint. Valójában a „reverse engineering" módszere megfelelő idő- és energiaráfordítás esetén lehetővé teszi, hogy miközben az adott protokollt (vagy a többi elemet) fekete doboznak tekintjük, és esetleg egyetlen sornyi kódot sem ismerünk belőle, attól még tökéletesen tudjuk szimulálni az inputra válaszként adott outputot. A Compaq annak idején teljesen törvényes módon így játszotta ki az IBM szabadalmaztatott BIOS-át, noha abból egyetlen sornyi kódot sem használt fel. Tizenöt programozó sok hónapi munkával és egymillió dolláros költségvetéssel az IBM-ével tökéletesen azonos módon működő BIOS-t állított elő. És a Linuxot nem is csak tizenöt programozó fejleszti!

A támogatás illúziója

A Linux, és általában a szabad forráskódú szoftver kontrájának esélye a Microsofttal szemben tehát ezen rekontra kilátásba helyezése ellenére is nyilvánvaló. Egy forráskód nélküli programmal — azon kívül, hogy lefuttatjuk és az vagy kielégítően működik, vagy nem —, keveset lehet kezdeni, de ha megvan a kód is, alapvetően más a helyzet. A számítástechnikához nem sokat értő főnök esetleg a „megbízhatóságot" fogja elsődlegesnek tartani, vagyis azt, hogy egy olyan cégtől ren-

delje meg a terméket (mondjuk a szerverszoftvert), amelynek nevével nap mint nap találkozik, a fejlesztők azonban nyilvánvalóan a hozzáférhető kódot fogják választani, mert abban saját maguk is megnézhetik, hogy mi hogyan és miért történik, és ha bármilyen probléma adódik, nyugodtan be is nyúlhatnak a motorházfedél alá, hogy szükség szerint kigazítsanak valamit.

Két teljesen különböző dolog, hogy mennyire hiszünk a Microsoft (vagy más cég) programozóinak csalhatatlanságában, és hogy mennyire szeretnénk fenntartani magunknak a változtatás és beavatkozás jogát. Ez utóbbi első megközelítésben csak elvi kérdés (ami szintén nagyon fontos!), de mélyebben elemezve kiderül, hogy messzemenő gyakorlati következményei vannak. Amikor a Linuxnál fény derült az ún. TearDrop IP attack hibára, akkor a párhuzamos debugging révén, vagyis annak köszönhetően, hogy egyszerre számos programozó nyúlhatott bele a kódba és kereshette a megoldást, kevesebb, mint 24 óra alatt megjelentek a javítások. Nincs az az óriáscég, amely ezt utánozni tudná. Legjobb esetben is meg kellene várni, hogy a gyártók kibocsássanak egy patch-et („befoltozót") vagy egy javított változatot. Az az érv tehát, hogy az NT alatt futó Microsoft szerverrel ellentétben az Apache mögött nem áll „support", igencsak gyenge lábakon áll. Éppen a „támogatottság" az egyik ok, ami miatt egy rendszergazda a szabad forráskódú megoldásra fog szavazni.

A „támogatási mód" eltérő felfogása már csak azért is kulcskérdés lehet, mert a számítógépesek összefogásával létrejövő szabadszoftverek mindig is modulárisan fognak felépülni, mert különben az egymástól függetlenül dolgozó programozók nem volnánának képesek fejleszteni. A Microsoft monolitikus stílusa nem csupán ellenszenves lehet a programozó társadalom nem beolvasztott rétege számára, hanem ilyen körülmények között hatékony működése is kérdéses, bármekkora gazdasági hatalom áll is mögötte. Annál is inkább, mert a modularitásnak sok egyéb előnye is van.

A Halloween Document szerzője maga is elismeri, az NT szerverekre a Linux egyebek közt azért jelent fenyegetést, mert *„a Linux az átlagos PC hardvert használja, és az OSS modularitiásából következően kisebb gépeken is működőképes, mint az NT. A Linuxot gyakorta alkalmazzák olyasmire, mint Domain Name Server üzemeltetése régi 486-osokon."*

A kód örök

Az OSS kihívására a megfelelelő válasz a Microsoft dokumentuma szerint legalább részben az lehet, hogy a mammutcég is megpróbálja átvenni az OSS által alkalmazott módszereket. Bár kérdéses, hogy a modularitásnál bevált eljárásokat mennyire lehet alkalmazni monolitikus építkezési mód esetén, nézzük meg ezeket a javaslatokat is.

Először is felmerülhet, hogy az NT forráskódjának licencét átengedjék egyes szervezeteknek, például bizonyos egyetemeknek és megbízható partnereknek a párhuzamos debugging (hibakeresés) előnyeinek kihasználására. Továbbá egyes kódrészleteket mindenki számára hozzáférhetővé tehetnének (ezáltal felkeltenék a hackerek érdeklődését) és olcsón vagy akár ingyen biztosítanának minden érdeklődő számára fejlesztőeszközöket.

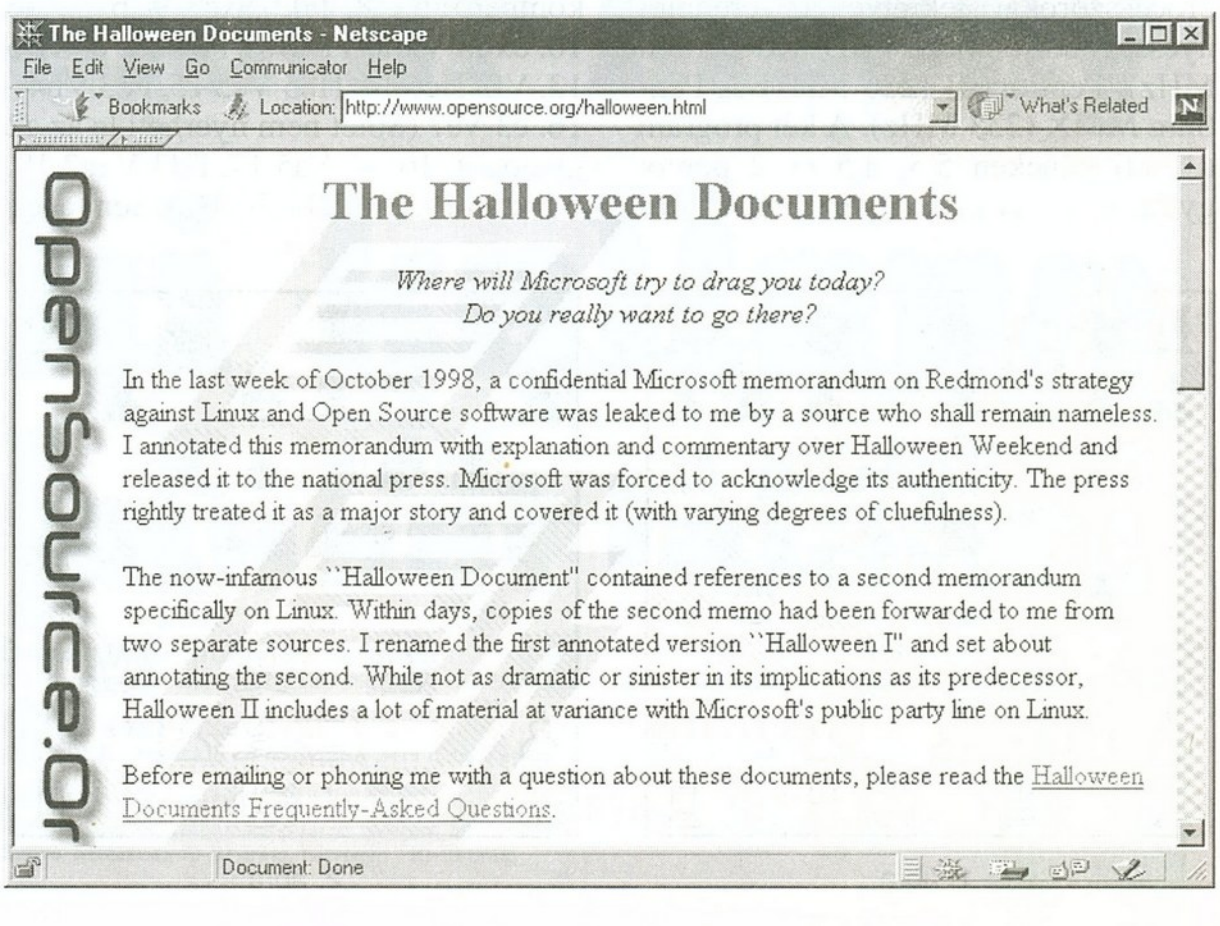
Elvégre a Linux is jelentős mértékben annak köszönhette sikerét, hogy a fejlesztőknek volt mivel dolgozniuk. Meg részben annak is, hogy a Linux feljesztői mindig is szerettek bíbelődni nem dokumentált API-kkal. Amennyiben a Microsoft néhányat, mint nem támogatottat közzé tenne, ezzel is növelné az NT iránti érdeklődést.

A felhasználók, illetve a számítógépes társadalom tájékozatlansága tehát nem minden esetben a legjobb üzlet. Egy nem céges programozókat tömörítő „tudásszféra" egészen biztosan hasznára válna a Microsoft ügyének is. *„Gondoljuk csak el* — írja a Halloween dokumentum szerzője —, *mi lenne, ha a microsoft.com egyik központi helyén a Visual Basic fejlesztőinek módjukban állna közzétenni teljes VB-projektjük forráskódját, és így megoszthatnák azt a többi fejlesztővel."* Nyilvánvalóan számos fejlesztő szeretné a világ legnagyobb szoftvergyártójának weblapján látni a nevét. És hasolóképpen az is nyilvánvaló, hogy a Microsoftnak emellett célravezető lenne *„az OSS hírcsoportok megfigyelése. Tanuljuk el az új ötleteket, és fogadjuk fel a legjobb, legragyogóbb elméket"* — olvasható a Halloween Documentben.

Ez utóbbi javaslat szintén hordozza a „birodalmi gondolkodásmód" elemeit, de mindent egybevetve lényegesen rokonszenvesebb, mint a protokollok kisajátítása és átírása. Persze az is elképzelhető, hogy a Microsoft mindkét módszert alkalmazza majd, vagy pedig annyira bízik erőfölényében, hogy egyáltalán nem tesz semmit, de az is lehet, hogy valami teljesen más dolgot talál ki. Nehéz, ha ugyan nem teljesen lehetetlen megjósolni, hogy mit hoz a jövő. Annyi bizonyosnak látszik, hogy a szabad forráskódú szoftver a Linux elterjedésével kinőtt a gyermekcipőből, és olyan tényezővé vált, amellyel a világ legnagyobb szoftvergyártójának is számolnia kell.

Ha a Linux egyhamar nem is szorítja ki az NT-t, meg a kereskedelmi forgalomban kapható, nem hozzáférhető forráskódú többi operációs rendszert, de a nagy gyártók mégis átvesznek bizonyos módszereket (például a párhuzamos debuggingot) az OSS-től, az mindenkinek csak hasznára válhat. És különben is: a programok meghalnak, de a kód örökké él. A Linuxé biztosan, a Windowsé talán.

Galántai Zoltán



Erőfelmérő verseny

Körmérkőzés mobil gépeken

A sakkprogramokról és a sakkszámítógépekről a figyelem egyre gyakrabban terelődik át a számítástechnikával támogatott sakkozásra, melynek során a gép és az ember játszma alatti együttműködése valósul meg. (Lásd erről szóló cikkünket 1998/10. számunkban.) Ugyanakkor a sakkszámítógépek és programok versenyeinek, mérkőzéseinek is számos új formája jön létre. A gépek paramétereinek javulása nagyobb játékerővel ruházza fel őket, amit persze próbára is kell tenni.

A számítógépes sakk bővülő versenyformáinak legutóbbi példája két élenjáró program, a német Fritz 5 és az angol Hiarcs 6 páros mérkőzése volt. A Magyar Sakkszövetségben megrendezett találkozó tizenkét partiból állt. A Portocom Kft három mobil számítógépet bocsátott hozzá rendelkezésre. A különböző paraméterekkel rendelkező gépek tudását a egyenlő esélyek biztosítása mellett kellett összemérni, hogy a hardver szerepét is reálisan lehessen megítélni.

Ezt a követelményt a szervezők új típusú versenyforma, egy tökéletesített körmérkőzés útján igyekeztek biztosítani: tizenkét partiban ugyanaz a két program váltakozó gépeken és változó színekkel vívott háromszor négy partiból álló mérkőzést egymással. Az összeredmény 8,5:3,5 volt Fritz 5 javára. A processzorokat tekintve az eredményesség sorrendje: AMD K6-2 (300 MHz), Pentium II (266 MHz) és Pentium MMX (233 MHz). A két program a fenti gépeken 5,5, 4,5 és 2 pontot gyűjtött be, vagyis egyértelmű a processzorsebesség és az eredményesség közötti korreláció. Egyébként mindhárom gép 64 MB RAM-mal volt felszerelve.

A játszmák a folyamatosan változó konfiguráció következtében változatosak, és ha nem is egyenletesen, de többségükben magas színvonalúak is voltak. Az alábbiakban bemutatunk egy érdekes partit, melyben Fritz 5 a legerősebb gépen világossal játszott.

Fritz 5.0 (K6-2/300) — Hiarcs 6.0 (PII/ 266)

Szicíliai védelem B 40.

1. e4 c5 2. Hf3 e6 3. d4 cxd4 4. Hxd4 Hf6 5. Hc3 Fb4 6. e5 Hd5 7. Vg4 Kf8 (1. ábra).

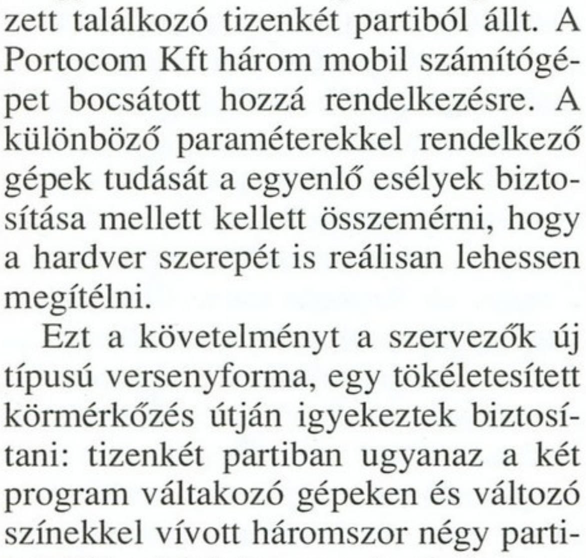
Igen éles változat, amelyben sötét a sáncolásról való lemondást világos vezérszárnyi gyalogjainak szétzilálásával kompenzálja. 8. Fd2 Fxc3 9. bxc3 d6 10. exd6 Vxd6 11. Bd1 Fd7 12. c4 Hf6 13. Vf3 Hc6 14. Hb5 Vc5 15. Fe3 Vb4+ 16. c3 Ve7 (Sötét nem nyerheti le az a2 gyalogot. 16. — Va5 17. Fd3 Vxa2 18. 0-0 után 19. Ba1 Vb2 20. Bfb1 fenyeget, 18. — Va5 19. Ba1 Vd8 20. Hxa7 után világos vezérszárnyi fölénye nyomasztó.) 17. Hd6 b6 18. c5! Hd5 19. Bd2 Bd8 (c3 sem üthető: 19. — Hxc3 20. Bc2 Hd5 21. cxb6 után c6-on lóg a huszár.) 20. Fe2 f6 21. Vg3 h5 (A h gyalog következő előretörését világos nem értékeli veszélyesnek, sötét számára viszont nem kínálkozik jobb ellenjáték.) (2. ábra.)

22. Hf5! (Fenyeget Hh4-g6.) 22. — exf5 23. Bxd5 h4 24. Vd6 h3 25. 0-0! (Ember számára nagy bátorság kellene ehhez a lépéshez, hiszen sötét betör a sáncállást védő gyalogok közé, és a h vonal is kinyílik. De a gép pontos számításai is kimutatják, hogy a h vonalon nem lehet a támadást folytatni.) 25. - hxg2 26. Bfd1 (3. ábra).

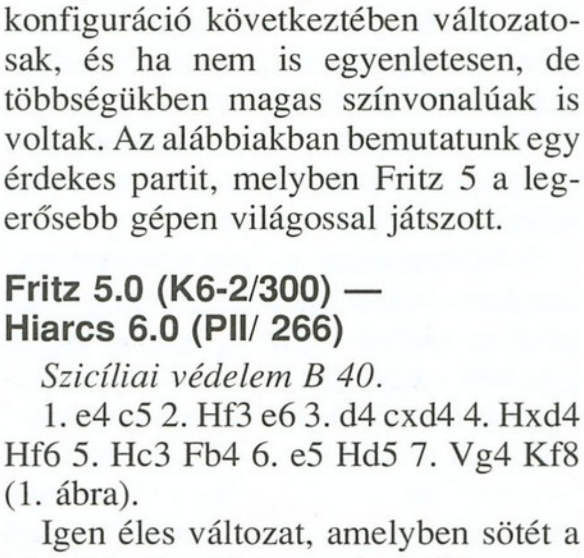
A d vonalon fokozódó nyomás alatt a sötét állás összeroppanni látszik. A vezér következő leadására érdekes taktikai közbeiktatást számít ki, ami azonban nem menti meg a helyzetet. 26. - Fe8 27. Vxc6! Ve4 (Erről van szó. 27. - Fxc6 Bxe8+ majd Bxh8 reménytelen, most viszont 28. Bxd8 nem megy, mert a vezér üt c6-on. Világosnak azonban döntő válasza van erre is.) 28. cxb6! („A legegyszerűbb lépések a legjobbak!" — mondja egy sakk-közmondás.) 28. - Fxc6 (Most sem kap eleget a vezérért, de mentsége nincs.) 29. Bxd8+ Kf7 30. Bxh8 f4 31. Fh5+ g6 32. Bh7+ Ke8 (4. ábra).

Erre még egy utolsó tréfa következik. 33. Be1! gxh5 34. Bxa7 Kd8 35. b7 Fxb7 36. Fb6+ és sötét feladta.

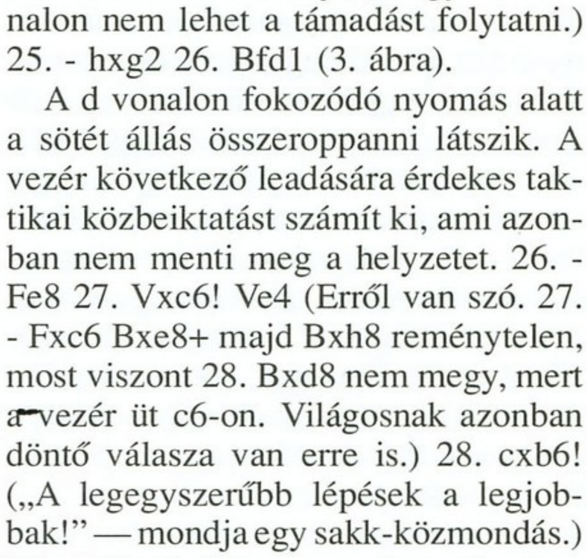
Lindner László



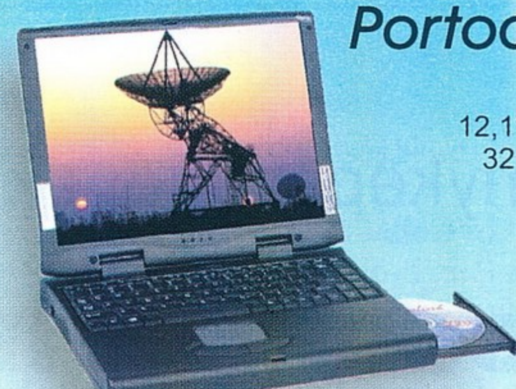
1. ábra

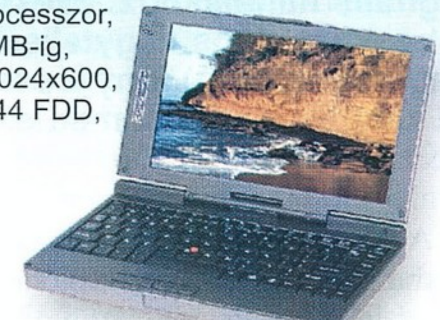


2. ábra



3. ábra





Digitális fényképezőgépek VI.

A csúcson: a hátfal

A digitális fotózásról szóló sorozatunkban alulról felfelé haladtunk, és most elérkeztünk a csúcshoz, a nagy felbontású digitális hátfalakhoz. Ezeket a digitális kiegészítőket a hagyományos nagyteljesítményű (középformátumú vagy nagyformátumú) fényképezőgépekre lehet ráilleszteni. (E kategóriát a Dicomed eszközei alapján mutatjuk be.)

A professzionális digitális fényképezésben két markánsan elkülönülő technikai irányzat létezik. Az egyiket jobb híján „szkenneres", a másikat „vakus" rendszernek nevezhetjük. A szkenneres eszközök a síkszkennerek elvén működnek, és expozíció közben CCD fényérzékelő sor mozog a film síkjában. Ilyen eszközök előállítási költsége kisebb, felbontása pedig nagyobb, mint a „vakus" megoldásé. Hátrányuk, hogy a megvilágítási idő hosszú, akár több perces is lehet, ezért csak mozdulatlan témák fényképezésére használható. Fényforrásként pedig vibrálásmentes, folyamatosan világító lámpákat kell használni.

A másik csoport kamerái sík CCD felülettel rendelkeznek, ezért pillanatfelvételre alkalmasak, és általában villanófény is használható hozzájuk, innen ered a vakus (flash) megjelölés.

A legtöbb vakus CCD pixelenként csak egy színösszetevő (kék, zöld vagy vörös) rögzítésére alkalmas. A teljes felbontású true color (RGB) képet utólag állítják elő szoftveres úton (interpolálással). Így megvan ugyan a teljes pixelszámnak megfelelő true color kép, de annak információtartalma csak egyharmadnyi ahhoz képest, mintha minden képpont mindhárom színösszetevőjét rögzítettük volna.



Dicomed BigShot

A Dicomed BigShot hátfalak a vakus kategóriába tartoznak. Felbontásuk 4260x4260, tehát több mint 16 millió képpont. Ezeknél az eszközöknél minden képpont tartalmazza a teljes színinformációt, mégpedig 3x12 bit (a BigShot 4000-nél 3x10 bit) színmélységben. A BigShot CCD felülete 6x6 cm, és az ekkora (közepes) filmformátumú Hasselblad, Mamiya RB stb. fényképezőgépek filmmagazinja helyére illeszthető. Három változatban készül, 1000, 3000 és 4000 típusmegjelöléssel. A BigShot 1000 elnevezésű változatban nincsenek színszűrők, így ez csak monokróm képek készítésére alkalmas. A 3000-es típusba vörös, zöld és kék színszűrőt építettek be, és a képet a kamera három, egymást automatikusan követő expozícióval készíti el. Fényforrásként megfelelő lámpa vagy villanófény is használható. (Ez utóbbival háromszor kell villantani.)

A 4000-es típusnál elegendő egyetlen expozíció is, a minimális expozíciós idő 1/300 másodperc. Ez a hátfal hangolható folyadékkristályos színszűrőt alkalmaz, ami lehetővé teszi a szűrő színének elektronikus úton történő gyors változtatását. Az 1/300 másodperces expozíciós idő alatt a pixelek egymás után rögzítik a három alapszínt. A színmélység alapszínenként 10 bit.

A BigShoot digitális hátfalak SCSI kábelen kommunikálnak a vezérlő szá-

Dicomed BigShot hátfalak technikai adatai

	1000	3000	4000
Maximális képfelbontás	4260 horizontális, 4200 vertikális képpont		
Színfelbontás, árnyalat	12 bit/képpont/szín		10 bit/képpont/szín
A képfájl maximális mérete	16 MB	48 MB	
Expozícios idő a maximális fájlméret exponálásakor	1/1000 sec	1/1000 (3x, < 10 sec)	1/300 sec
Formátum	61x61 mm (Hasselblad), 4x5" (96/Q2)		
Aktív terület	Teljes méretű: 61x61 mm		
Érzékenység	A CCD lap érzékenysége (800 ASA)		
Támogatott fényforrások	Tungsten, HMI, high frequency fluoroscent, napfény		
Kamera adapterek	Hasselblad 553 ELX, (500 EL, ELM)		
Számítógép követelmény	PowerMac 64 MB RAM, SCSI-3		
Működési elv	Hangolható folyadékkristályos szűrő (LCTF)		



mítógéppel, a mobil kivitel 1 GB-os Jaz drive egységet tartalmaz.

Dicomed Pro széria

A Dicomed Pro hátfalak a műszaki fényképezőgépek 100x125 mm méretű síkfilm kazettájának helyére illeszthetők. Az érzékelő aktív felülete 72x90 mm, ami 115 mm-es képátlót jelent. Ezek az eszközök szkenner rendszerűek, vagyis egy CCD-sor halad el az objektív által kirajzolt kép síkjában. A teljes felület letapogatása 27 másodperctől 15 percig terjedő időt vesz igénybe, de ha nem kívánunk maximális felbontású fájlt készíteni, az expozíció ideje lerövidíthető. A hosszabb expozícióért viszont rendkívül nagy felbontást kapunk cserébe. A keletkező kép 6000x7520, azaz több mint 45 millió képpontból áll. A kiolvasott színmélység 3x12, vagyis 36 bit. Ez a hagyományos 16 millió színárnyalat helyett 65 milliárd árnyalatot jelent.

A grafikai programok általában nem tudnak 24 bitnél nagyobb színmélységgel dolgozni, ezért a Dicomed Pro hátfalak vezérlő elektronikája egy speciális DSP célprocesszort tartalmaz, amely átalakítja a 36 bites képet 24 bitessé röptiben („on-the-fly" üzemmódban), tehát várakozás nélkül. A jelleggörbét, amely alapján a DSP processzor az átalakítást végzi, a vezérlő számítógépen gafikusan vagy numerikusan meg lehet adni. Így eldönthető, hogy a beolvasott kép melyik része vagy tónusa hordozzon több részletet a végső 24 bites képen. A letapogatott kép információtartalma kb. 200 MB, amely a 24 bitesre történő átalakítás után 132 MB-ra redukálódik. A Dicomed Pro sorozat lényegében egyetlen kamerahátfal három különböző kiépítettségben és különböző szolgáltatásokkal.

Studio Pro. A legolcsóbb, „belépő" verzió. Ennél a változatnál a felbontás szoftveresen van korlátozva 3000x3760 képpontra. Megfelelő másik szoftverrel a következő osztályba, a Studio Pro XL-re lehet átlépni, tehát az optikai érzékelő már az olcsóbb modellben is azonos. A Studio Pro XL a teljes 6000x7520-as felbontást produkálja. Ezek a rendszerek nem tartalmaznak vezérlőegységet és adathordozót, mert csak a megfelelő nagyteljesítményű számítógéppel összekapcsolva használhatók. Mind a két változat hálózati áramot igényel, tehát stúdiókörülményekre vannak felkészítve.

Field Pro. Ez a típus elektromos hálózattól függetlenül, külső helyszínen is működik, amit a rendszerhez tartozó akkumulátor, a hordozható vezérlőegység és a hordtáska tesz lehetővé. A vezérlőegység DSP processzort és egy 1 GB-os merevlemezes tárolót tartalmaz. Emellett szükséges hozzá egy hordozható számítógép, amelyen a rendszer kezelőprogramja fut. Használható a Field Pro természetesen stúdióban, asztali géppel is, ilyenkor a vezérlés a számítógép soros kommunikációs vonalán keresztül történik, az adatok pedig SCSI interfészen keresztül kerülnek rögzítésre a winchesteren.

Illusztrációként elhelyeztük CD-mellékletünkön egy Dicomed BigShot 3000 hátfallal készült felvétel két képállományát. A DEMO-1.TIF képen a trljrd (eredetileg 24,8 MB-os) felvétel látható 20%-osra lekicsinyítve. Ezen bejelöltük azt a részletet, amelyet a DEMO-2.TIF fájl tartalmaz kicsinyítés nélkül, eredeti felbontásban. Ebből jól megítélhető a felvétel minősége.

Dékán István

A Dicomed Pro család technikai adatai

	Studio Pro	Studio Pro XL	Field Pro
Maximális képfelbontás (képpont)	3000x3760	6000x7520	
Színfelbontás, árnyalat	12 bit/képpont/szín, 36 bit/képpont, 16 millió képpontú fájl		
Minimális és maximális képfájl	2–32 MB	2–129 MB	
Expozícios idő preview készítésekor	–		28 sec
Expozícios idő maximális fájlméretnél	75 sec	150 sec	
Formátum	4x5 inch teljes méret (a Polaroid kazetta helyére)		
Aktív terület	72 x 90 mm (115 mm átló)		
Érzékenység	100–1600 ASA		
Támogatott fényforrások	Villogásmentes, Tungsten, HMI, high frequency fluoroscent, napfény		
Szkennelési sebesség	1/40 – 1/8 sor/sec		
Alkalmazott érzékelőtechnológia	Trilinear Array Capture Technology, egymenetes olvasás		
Számítógépes hardverkövetelmény	PC / PowerMac 64 MB RAM, SCSI-3		Mac Powerbook
Egyéb jellemzők	Stúdió model		Hordozható
	Dual SCSI, 1 GB belső HDD, soros kamerakontroll		

Calmira 3.01

Továbbra is 16 biten

A Windows 98 megjelenésével a 16 bites Windows (és a Calmira) kitartó hívei újabb választás elé kerültek: álljanak-e át most a 32 bites platformra a Windows 95-ös lépcsőt immár végérvényesen kihagyva, vagy továbbra is kitartsanak a sokak által egyre elavultabbnak tartott operációs rendszerek és alkalmazások mellett.

Három évvel ezelőtt igencsak nehéz döntést kellett hozniuk a felhasználóknak, ha úgy határoztak, hogy maradnak a régi, 16 bites rendszer mellett. Most egyszerűbb és könnyebb a helyzetük, mert a Windows 98 messze nem jelent akkora kihívást az előző verzióhoz képest, mint amekkorát annak idején a Windows 95 jelentett. Persze nem állítható, hogy például az USB támogatás, a FAT32, az aktív desktop ne lenne pozitív előrelépés, de a 386/486 kategóriájú számítógépekkel rendelkezők most mégis könnyebb szívvel vállalhatják a régi, 16 bites környezetet. Továbbra sem kell teljes számítógépcserét vállalniuk az új operációs rendszer kedvéért. És az sem mellékes, hogy a hardver beruházási költségeihez képest a jogtiszta szoftver beszerzése ma jóval nagyobb tétel, mint amekkora az előző operációsrendszer-váltáskor volt.

A továbbra is 16 bites Windowst használók döntését az a tény is megkönnyítette, hogy a Calmirát továbbfejlesztették, megjelent a 3.0-s verzió, melyet szinte azonnal követett a javított 3.01-es változat. Az sem véletlen, hogy a Calmira teljes verziószámot váltott. Amíg ugyanis a 2.1 és a 2.2 verzió csak kisebb változtatásokat és javításokat tartalmazott, a 3.0-s változat fontos új tulajdonságokkal rendelkezik. Ennek magyarázata, hogy az eddigi egyszemélyes fejlesztés kiegészítésére, a további munka összefogására és koordinálására fejlesztőcsoport alakult, és Calmira weblapot is működtetnek. A verziószám mellett ezentúl Calmira II megkülönböztető névvel is ellátják a programot.

Az újdonságok

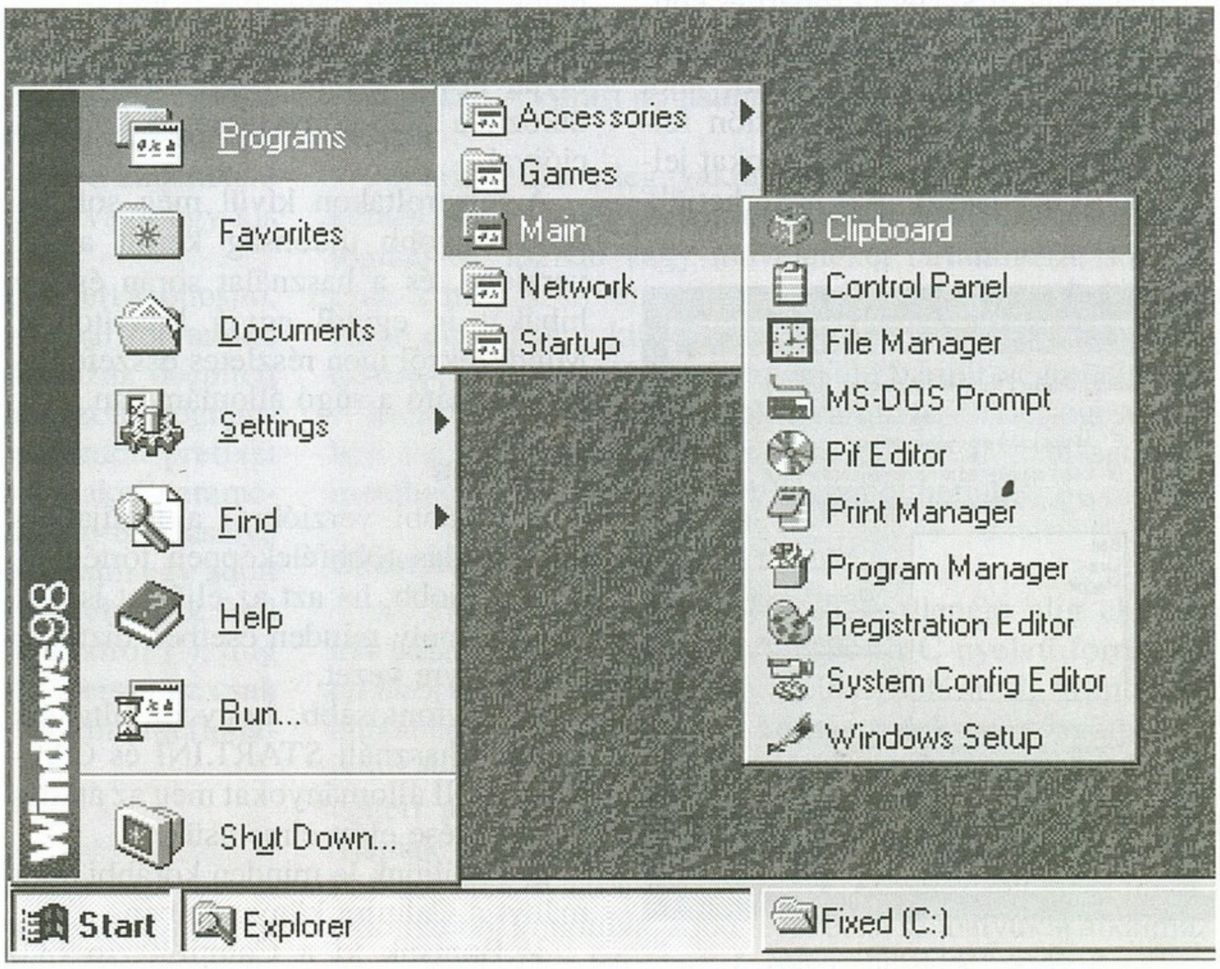
A fejlesztőket láthatóan két fő cél vezérelte: egyrészt a 32 bites Windows felhasználói felületéhez még jobban hasonlító külső megjelenés, másrészt a felhasználók tapasztalatainak és javaslatainak beépítése a minél könnyebb és rugalmasabb kezelés érdekében. Az első szempontnak most már tökéletesen megfelel a Calmira, hiszen komplett új ikonkészlettel, bitmapekkel, Start menüvel, kartotékos dialógusokkal, nyomógombokkal tökéletesen imitálja a nagyobb testvér küllemét. Csupán az ablakok fejléce és a Start menü használatának némileg eltérő módja utal rá, hogy mégsem a Windows 98-cal van dolgunk. De amint 1. képünk is mutatja, meglephetjük azokat, akik kitartóan megpróbálnak bennünket rábeszélni a Windows 98-ra. És persze megemlíthetők a szinte tökéletes illúziót szolgáló kuriózumok, mint például a kilépéskor megjelenő dialógus, amely elsötétíti a hátteret, valamint a Windows 95-ből kimásolt, és a programbefejezés tényét jelző háttér, amely tetszés szerint cserélhető is.

Nézzük át most azokat az újdonságokat, amelyek a felhasználókkal való folyamatos kapcsolattartás révén kerülhettek be a legújabb Calmira verzióba:

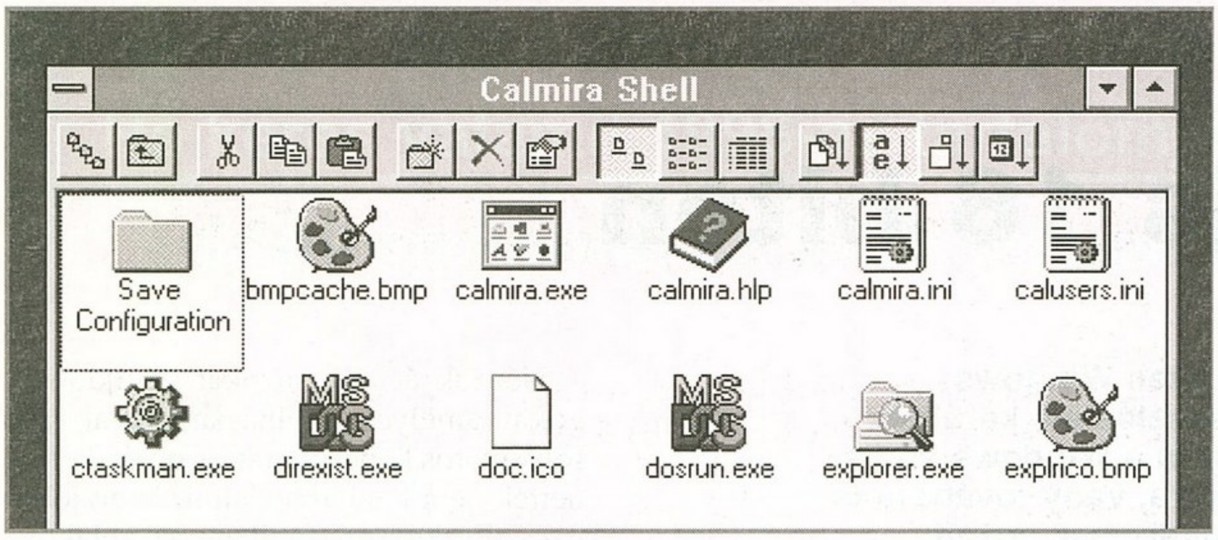
— Eszközsor (toolbar) az ablakokban (2. ábra). Nem tűnik különlegesnek, de nagyon sok felesleges egérkattintgatást megtakaríthatunk vele. Válthatunk az ablak tartalmának megjelenítési formái között (ikon és kétfajta lista), továbbá a tulajdonságlap és a törlés is elérhető egyetlen kattintással. Ott van továbbá a szülőkönyvtár kiválasztásának lehetősége is. Mindez apróság, de a mindennapi munkát igen meggyorsítja.

— Shell DDE. Ez valóban nagyon hiányzott a 2.0-ból. Azok az installáló programok, amelyek új csoportablakokat hoztak létre, és új bejegyzéseket tettek a Program Kezelőben, ezentúl módosíthatják a Start menü megfelelő bejegyzéseit, illetve új elemeket vehetnek fel a Start menübe.

— Inkrementális keresés az ikonablakokban: a '.' leütése után töredék névkezdet megadásával is gyorsan kereshetünk az ikonok és állományok között.



1. ábra. Start menü Windows 98 mintára



2. ábra. Eszközsorral ellátott ikonablak

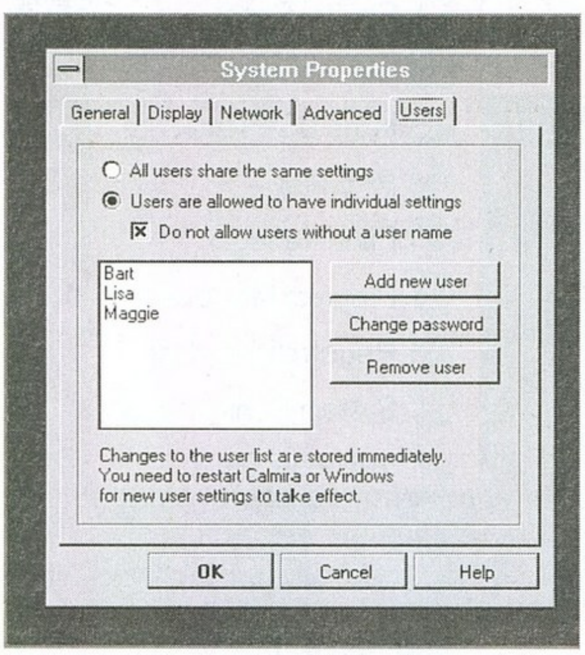
— Az ablakok címsora most már a directoryhoz tartozó hosszú nevet jeleníti meg.

— Beállítható a hosszú fájlneveket tartalmazó állomány neve is.

— A taskbar (tálca) a Calmira integráns része lett, jobban konfigurálható, akár a képernyő tetejére is tehető. Az innen hívható rendszermenü az új verzióban valóban a illető program rendszermenüjét jeleníti meg, így láthatók azok a menüpontok is, melyeket esetleg a program illesztett a rendszermenühöz.

— Saját taskmanager (kezelő). A CTASKMAN.EXE használható az eredeti kezelő kiváltására, így a Ctrl+Esc megnyomásakor ez a kezelő fog megjelenni. Ehhez a SYSTEM.INI fájl [Boot] szekciójába tegyük bele a taskman.exe=c:\calmira\ctaskman.exe sort.

— A legérdekesebb újdonság — és ezt hagytam utoljára — a Calmira többfelhasználós környezetben való alkalmazhatósága. A System Properties konfiguráló dialógusa ugyanis kiegészült egy új, Users lappal (3. ábra), ahol lehetőségünk van rá, hogy külön felhasználókat vegyünk fel, és azokat jelszavakkal is ellássuk. Azt is kérhetjük, hogy felhasználói név és jelszó megadása nélkül a Calmira ne engedje meg a hozzáférést. Letilthatjuk, hogy az egyes felhasználók a Calmira konfigurációjában változtatásokat hajtsanak végre. Minden egyes felhasználónak külön Start menüt készíthetünk, és persze külön konfigurációs paramétereket állíthatunk be, így lehetővé válik, hogy például iskolai vagy könyvtári környezetben az egyes felhasználók más és más felülettel és lehetőségekkel találkozzanak, anélkül, hogy vétlen vagy szándékos károkozásuktól tartanunk kellene. Ez persze nem azonos értékű a Windows NT által biztosított védelmi lehetőségekkel, de a Calmira azért készült, hogy akár 386-os processzorral felszerelt számítógépen is futhasson.



3. ábra. Többfelhasználós környezet kialakítása

A felhasználói név maximum nyolc karakter hosszúságú lehet. Mivel a Calmira a felhasználók által látott konfigurációt a Profiles könyvtárnak a felhasználók neveivel megegyező alkönyvtáraiban tárolja, ügyeljünk arra, hogy olyan nevet adjunk meg, amely nem ütközik a DOS könyvtárnevek konvencióival.

A felsoroltakon kívül még sok kisebb-nagyobb újdonság került az új verzióba, és a használat során észlelt hibákat is egytől egyig kijavították. Mindezekről igen részletes összefoglaló olvasható a súgó állományban.

Telepítés

A korábbi verziókról a legújabbra való átállás többféleképpen történhet, ezért legjobb, ha azt az eljárást ismertetjük, amely minden esetben biztosan eredményre vezet.

1. Legfontosabb, hogy az általunk jelenleg használt START.INI és CALMIRA.INI állományokat még az átállás megkezdése előtt elmentsük.

2. Töröljünk le minden korábbi állományt a Calmira könyvtárában.

3. Bontsuk ki a Calmii30.zip (3.0 teljes készlet), majd a 301patch.zip (3.01 javító készlet) fájlokat a Calmira könyvtárban, ebben a sorrendben.

4. A Cii30src.zip (3.0 forrás) és a 301src.zip (3.01 javítások forrásai) állományokat bontsuk ki abban a könyvtárban, ahol a forrásokat tartjuk.

5. Ha szeretnénk a Windows 95-öt hűen utánzó felületet használni, akkor a w95theme.zip állományra is szükségünk lesz. Ezt egy átmeneti könyvtárban bontsuk ki. Másoljuk a w95icons.dll-t a Calmira könyvtárba. A banner*.bmp fájlok közül válasszuk ki azt, amelyiket a Start menü oldalsávjának szánjuk, és ezt banner.bmp néven másoljuk a Calmira könyvtárába. A többi bitmap állományt is másoljuk ide.

6. Ha a 16x16-os ikonok helyett a 32x32-es ikonokat szeretnénk használni, módosítanunk kell a CALMIRA.INI fájlt: az [Icons] szekcióban cseréljük ki az ikon indexeket a sorvégi megjegyzésekben található indexekre.

7. Írjuk felül az új Ctl3dv2.dll-lel a régi állományt, ügyeljünk arra, hogy csak egyetlen ilyen állomány legyen a Windows által elérhető helyen.

8. Ezután kezdődhet az újrakonfigurálás, mivel az ini fájlok most az alapbeállításokat tartalmazzák. Kétféle megoldást választhatunk. Az egyik esetben a Calmira beállító dialógusait sorra véve állítsuk vissza korábbi beállításainkat, és persze ha szükséges, módosítsuk az új opciókat is, majd importáljuk újra a Program Kezelő programcsoportjait. A másik módszer nagyobb figyelmet követel: nyissuk meg egy szövegszerkesztőben az ini fájlok korábbi és jelenlegi verzióit, majd a szükséges sorokat emeljük át a régi állományokból az újakba. Vigyázzunk arra, hogy ugyanaz a sor mindig csak egyszer szerepeljen.

9. Ezután indítsuk el a Windowst, majd ellenőrizzük, hogy a beállítások megfelelőek-e, és hogy a Calmira külleme is olyan-e, amilyet szeretnénk.

10. Végül készítsünk mentést az ini fájlokról. Most már három ilyen fájl van, és ha többfelhasználós környezetet is kialakítottunk, akkor a Profiles könyvtárban is újabb ini fájlok keletkeznek majd, melyeket időnként ugyancsak érdemes elmentenünk.

A Calmira fejlesztőit a www.calmira.org Internet címen érhetjük el, innen néhány további kiegészítő is letölthető, amellett, hogy véleményünket, ötleteinket is elküldhetjük a szerzőknek. Ugyancsak innen érhető el az eddig elkészült öt nemzeti verzió is (sajnos magyar nyelvű Calmira még nincs köztük).

Szűcs János

GNU for DOS — XII.

HTML makró — haladóknak

A ma használatos programnyelvek egyre inkább megengedik a makrózást, tehát egyre kevésbé van szükség külső makrónyelvre. Már el is döntöttem, hogy nem érdemes foglalkoznom az m4 programmal, amikor kezembe akadt a Linux Gazette 22. száma, amelyben szerepel, hogy miként alkalmazható az m4 a HTML oldalak szerkesztésére. Ez annyira frappáns írás volt, és olyan jó alkalmazást talált az m4-re, hogy végül is nem tudtam ellenállni a kísértésnek, hogy mégis írjak az m4-ről.

A programok írásakor jelentős könnyebbséget jelent, ha konstansokat használhatunk, mert ha a programban szereplő adat mérete megváltozik, elegendő egy helyen elvégezni a módosítást, és nem kell az egész programot végigtallózni a megfelelő értékekért. A C nyelv még azt is megengedi, hogy a programszöveg egyes részeit bizonyos feltételek esetén kihagyja a fordításból, de lehetőségünk van más forrásfájlokat is beemelni a fordítandó szövegbe. Mivel Unix környezetben alapvető a C rendszer és vele együtt a preprocesszor (előfeldolgozó) jelenléte, így nem csoda, hogy azt más programok is felhasználják, mint például a 'calendar' is. Egyes helyeken addig mennek a C előfeldolgozójának kiokosításában, hogy a rövidke Basic programokat helyes kis C programként fogadja el a fordító.

Egy kis bemelegítés

Minthogy a C előfeldolgozójának igen korlátozottak a lehetőségei, készítettek egy jobban kihasználható előfeldolgozót is. Ez az m4. Sokan használják ezt a programot, pedig azt sem tudják, hogy a gépükön van. Ez azért történhet meg, mert az 'autoconf' programcsomag, amely az adott gép beállításainak megfelelően készíti el a Makefile-t, alkalmazza ezt a programot.

Akik nem programoznak, csak annyit kell megjegyezniük, hogy ez a program szűrőként dolgozik, azaz a bemenetére érkező szöveget kisebb-nagyobb átalakítás után a kimenetére írja. Ez az átalakítás lehet szöveghelyettesítés vagy egy kis aritmetika is.

A makró neve lehet a programnyelvekből ismert azonosító, azaz betűvel kezdődő, majd betűvel és számmal folytatódó karaktersorozat. Van néhány előre definiált, azaz beépített makró. Ha ez véletlenül egybeesne az alkalmazott programnyelv kulcsszavaival, akkor egy általunk megadott prefixet ragaszthatunk minden beépített makró elé. A makró paramétereit a makró neve utáni zárójelben, vesszővel elválasztva adhatjuk meg. Ha nem szeretnénk, hogy a program egy adott karaktersorozatot kifejtsen, akkor azt idézőjelek közé kell tennünk, ahol a kezdő idézőjel a fordított aposztróf (‘), míg a záró idézőjel a hagyományos aposztróf ('). Persze ez csak alapállapotban van így, kényünk-kedvünk szerint átdefiniálhatunk mindent!

Lássunk egy egyszerű definíciót:

```
define(‘Szia',‘Udvozollek dicso lovag!')
Szia!
```

Ezt a példát kipróbálva láthatjuk, hogy a köszöntés végén két felkiáltójel szerepel, az elsőt a makrókifejtésből kaptuk, a második pedig érintetlenül maradt, tehát óvatosan kell bánni a makrók definiálásával! A makrókat, miután megtették kötelességüket, törölhetjük. Ha viszont csak ideiglenesen akarunk a makrónak más jelentést adni, a 'pushdef' és a 'popdef' segítségével könnyedén lépkedhetünk a jelentések között. A makróparaméterekre az AWK vagy az sh-shell szintaxisában hivatkozhatunk, például:

```
define(‘masodik',‘$2')
masodik(1,2,3,4,5,6,7)
```

A Forth szerelmesei örömmel fedezhetik fel, hogy itt is lehetőség van a beépített makrók újradefiniálására, ám még ekkor is elérhetjük az eredeti definíciókat, ha kicsit körülményesen is.

Részletes „beetetés"

Az m4-ben megvalósítható a feltételes makródefiníció, és mivel van benne elágazás, ciklus és rekurzió, akár még a for-ciklust is használhatjuk. Mivel ilyen makrók írása a jobbaknak sem egyértelmű, az m4 fel lett vértezve hibakeresésben segítő makrókkal.

Vannak függvények, amelyek megadják, hogy egy minta (esetleg reguláris kifejezés) hol szerepel a megadott szövegben. Lehetőségünk van karakter-előfordulásokat lecseréltetni, így például törölhetjük a kisbetűket, vagy nagybetűsíthetjük azokat. A kiírást pofásíthatjuk a C-beli printf-hez hasonló módon.

Az aritmetika a leggyakrabban használt növelés és csökkentés mellett a négy alapműveletet, hatványozást, forgatást, logikai és, logikai vagy műveleteket tartalmazza, sőt nem csak a tízes számrendszert használhatjuk. Ha mindez nem lenne elég, akkor külső programokat is meghívhatunk, ám ez nemcsak számolást, hanem szinte bármit végrehajthat.

Méretesebb m4 programok újrafuttatása feleslegesen terheli a számítógépet, ezért a program belső ábrázolását fájlba menthetjük, és onnan később visszatölthetjük.

Személyes viszonyulás

Az előbbi bámulatos lehetőségek ellenére alig akadtam m4 alkalmazásra. Csupán a ThisForth C nyelvű forrásában találkoztam az m4 komoly felhasználásával. Miután én általában a TeX segítségével készítem dokumentumaimat, én is inkább a texpp előfeldolgozót használom, mert az jobban ismeri a TeX-et, mint bármi más előfeldolgozó program. Vannak viszont olyan esetek is, amikor az m4 jól jöhet. Például igen gyakran szedek ki levezetéseket, ahol bizonyos axiómákból kiindulva, adott levezetési szabályokat alkalmazva kell eljutni a kérdéses formuláig. A probléma ezzel az, hogy az egész levezetést csak visszafelé tudjuk megcsinálni,

és úgy kell hivatkozni a korábbi sorokra, hogy nem ismerjük, végül milyen sorszámot kapnak. Egy lehetséges megoldás, hogy miközben visszafelé írjuk a levezetést, minden sor kap egy elnevezést, amely egy makró neve lesz. Ha teljesen elkészült a levezetés, akkor már tudjuk, hogy az egyes makrónevekhez milyen sorszám tartozik.

Mivel a levezetésben szereplő formulák igen gyakran ismétlődnek, és a formulák begépelése nem a legegyszerűbb feladat, ezért a formulákat makrókkal érdemes helyettesíteni. A kevesebb gépelés mellett ennek előnye, hogy kisebb a tévesztés esélye, és ha ugyanaz a rész a levezetésben egy-két formula eltéréssel megismétlődik (ami nem ritkaság), akkor a levezetés adott részét is lehet makróval helyettesíteni.

HTML oldalak szerkeszetése

Talán nem mindenki tudja, hogy minden HTML oldal egy sima szövegfájl, amelynek a szerkezetét és paramétereit kacsacsőrös zárójelek közé tett kulcsszavak (címkék) jelölik ki. Valaha ezeket a kulcsszavakat úgy alakították ki, hogy a fájlok mérete minimális legyen, így még a lassú vonalakon is normális idő alatt letöltődjenek. Azután a látvány győzött az ésszerűség felett, új kulcsszavak jelentek meg, böngészőnként különböző változatok alakultak ki. A speciális grafikus szerkesztők pedig vagy az egyik, vagy a másik böngészőt részesítik előnyben. Ezekkel a szerkesztőkkel azután olyan oldalak is születhetnek, amelyek saját böngészőnkön remekül néznek ki, egy másikon azonban „megcsúnyulnak", vagy meg sem jelennek.

Az pedig már külön történet, hogy a különféle konverterekkel előállított oldalak mérete esetleg nagyságrenddel nagyobb, mintha direkt HTML-re írtuk volna meg. Eljutottunk oda, mint a könyvkiadással: aki tudja, hogy melyik billentyű merre található, az már képes könyvet készíteni. Most aki tud kattintgatni az egérrel, az WWW oldalakat készíthet. Egy oldalt még nem is nehéz megcsinálni, de amikor már negyven vagy ötven oldalt kell folyamatosan karbantartani, akkor kiderül, hogy a rest kétszer (vagy jóval többször) fárad.

Az a tény, hogy egy HTML oldal egyszerű szövegfájl, egyrészt azt jelenti, hogy a legegyszerűbb szövegszerkesztővel is megírhatjuk, másrészt azt, hogy felhasználhatjuk azokat a segédprogramokat, amelyek a szövegfájlok feldolgozására szolgálnak. Megemlíthetjük például az RCS-t, amellyel az oldalak verzióit tárolhatjuk, kereshetjük vissza, és összefésülhetünk bizonyos változtatásokat. Korábban ezeket az oldalakat a 'sed' program segítségével készítettem. Nem sokat használtam ki a sed lehetőségeiből, de így is rengeteg ismétlődő feladattól szabadultam meg.

A jelenleg elterjedt HTML nem engedi meg makrózást, pedig igencsak jól jönne. Például a linkek megadása kicsit unalmassá válik, ha ügyelnünk kell arra, hogy az oldal nyomtatott változata is ugyanannyi információt tartalmazzon, mint a képernyőn olvasható. Gondoljunk csak arra, hogy milyen sok távoli link elnevezése a „here", vagy magyar változata, az „itt". Ha rákattintok, akkor persze elolvashatom, de ha a kinyomtatott papírt előszedve akarom később felkeresni az adott címet, akkor lehet, hogy a kinyomtatott oldal már rég megszűnt, míg ahova a link mutat, az még létezik, de én innen már el nem érhetem. Erre ajánlják, hogy a link tartalmazza a címét! Ekkor viszont két példányban kellene minden címet begépelni, ami egy idő után kezd unalmassá válni. Nem lenne egyszerűbb a

```
_LINK('http://www.alaplap.hu')
```

alkalmazása a

```
<a href="http://www.alaplap.hu">http://www.alaplap.hu</a>
```

helyett? Ami engem igazán untat, az az oldal alján szereplő — nálam mindig ugyanazt az információt tartalmazó — rész begépelése. Az itt szereplő szöveget már a seddel is egy külső fájlból olvastam be, viszont az aktuális dátum is szerepel az adatok között, így ezt a segédfájlt időről időre át kellett írnom. Mára ez már így néz ki:

```
_STB
```

Azt hiszem, ezen nagyon sokat már nem is egyszerűsíthetek. Lássuk, mit is csinál ez:

```
define('_STB', '<ADDRESS> <CENTER>
Utolso valtoztatas idopontja:' esyscmd(date +%Y/%m/%d)
'<BR>
Aszalos Laszlo (aszalos@math.klte.hu)
</CENTER>
</ADDRESS>
</BODY>
</HTML>'
)
```

Mindezt beszoríthattam volna két sorba is, de így jobban átlátható, hogy miről is van szó. A date külső parancs jól paraméterezhető Unix alatt, DOS alkalmazásakor viszont az m4-et kell rábírni, hogy használható alakot készítsen a DATE parancs kimenetéből. Ha a hónap nevét magyarul szeretnénk megjeleníteni, akkor is az m4 segíthet a feltételes szerkezettel.

Egyebek

Vannak, akik ugyanazokat az oldalakat több helyre is felrakják, kisebb változtatással. Ekkor is hasznos lehet az m4, mert az URL megfelelő részét egy makróra cserélve pillanatok alatt átköltöztethetjük az oldalakat, függetlenül attól, hogy húsz oldalunk van, vagy pedig kétszáz.

Gyakran előfordul, hogy oldalaink egyetlen sort alkotnak, mint például a könyvben a fejezetek. Ekkor szokásos az oldal elején vagy a végén a következő linkeket feltüntetni: első, előző, következő, utolsó. Egyszer még begépeli az ember, de ha valamikor majd ikonokat is szeretne hozzájuk kapcsolni, kínkeserves lesz a változtatás. Pedig elvileg csak két adatban tér el minden ilyen bejegyzés, az előző és a következő oldal címében, tehát érdemes használni az alábbit:

```
define('GOMBSOR','
<a href="kezd.html">elso</a>
<a href="$1">elozo</a>
<a href="$2">kovetkezo</a>
<a href="vege.html">utolso</a>')
```

És ezt a következőképpen használjuk fel:

```
GOMBSOR(err13.html,err15.html)
```

Ha már minden remekül működik, és a határidő sem sürget, akkor a GOMBSOR definícióját lehet finomítgatni, csinosítgatni.

A kiadványszerkesztés, így a WWW lapok szerkesztése is több, logikailag elkülönülő részből tevődik össze, és komolyabb helyeken az egyes munkafolyamatokat más és más csinálja. Ha a látványért felelős meggondolja magát, és bizonyos kiemelést más karakterkészlettel akar megjeleníteni, akkor ez nem jelenti azt, hogy a szerzőnek minden ilyen címkét át kell írnia. Léteznek finomabb megoldások is, például lehet definiálni _DOLT, _VASTAG, _IDEZET és hasonló makrókat, amelyek definícióját a látványmester véglegesíti. A szöveg megírása folyamán elfogadható mondjuk a

```
define('_IDEZET',<i><b>$1</b></i>)
```

definíció is, azután ha minden szöveg kész, jöhet a cicoma.

Hosszabb művek esetén előfordulhat, hogy megváltozik az eredetileg elképzelt szerkezet, egy alfejezet kinövi magát, és fejezet lesz belőle, vagy egy fejezetnek szánt rész csak

alfejezetnyi anyagot tartalmaz. Ezért nem érdemes a számozást kőbe vésni. A listáknál egyszerű a helyzet, ott a legeslegutolsó pillanatban képződik a sorszám. Fejezeteknél nincs ilyen lehetőség, használhatjuk viszont a következőket:

```
define(‘_FEJEZET’,0)
define(‘_H’,‘define(‘_FEJEZET’,incr(_FEJEZET))
   <H2>_FEJEZET. $1</H2>’)
```

Majd:

```
_H(Bevezeto) bla bla bla...
_H(A viragok es a mehecskek) bla bla bla...
```

Ezután semmi probléma nem lesz, ha két fejezetcím felcserélődik, vagy újabbat szúrunk be. Sokan szeretik, ha a hosszabb oldal egy tartalomjegyzékkel kezdődik, és egyből a kívánt részre lehet ugrani. Ezt több módon is meg lehet oldani. A legegyszerűbb módszer, hogy az irομány elején minden fejezetcím részére egy rövidítést (makrót) készítünk, és később majd ezt használjuk fel:

```
define(‘_UGRAS’,<a href="‘#’$1">$1</a>)
define(‘_CIM’,<a name="$1"><H2>$1</H2></a>)
define(‘_BEVEZ’,‘Mi is van ezen az oldalon?’)
define(‘_MIERT’,‘Miert is irtam ezt az oldalt?’)
...
<UL><LI>_UGRAS(_BEVEZ)
   <LI>_UGRAS(_MIERT)
   ... </UL>
_CIM(_BEVEZ) bla bla bla ...
_CIM(_MIERT) ...
```

Persze ezt és az előző makrókat össze lehet kapcsolni, al- és al-alfejezeteket is készíthetünk automatikus számozással. Jóval bonyolultabb makrókkal és megfelelő Makefile-lal elérhetjük, hogy csak a tartalomjegyzék helyét kell megadni, és ide automatikusan bekerülnek a fejezetcímek.

„Intelem"

Mindezt még lehetne fokozni, de programunk bonyolultsága pillanatok alatt túllépne felfogóképességünkön. Ezért inkább kis lépésekben ismerkedjünk az m4-gyel, és csak ott alkalmazzuk, ahol valóban könnyít a munkánkban. Ne felejtsük el, hogy ez csak az egyik eszköz a sok közül!

A CD-mellékletre felkerült a program forrása, a DOS-os futtatható fájl és egy halom példaprogram. Az utóbbiak egy része a SimTel archivum win3/util alkönyvtárából, a u16_v42.zip fájlból származik. Aki alaposabban is meg szeretne ismerkedni az m4-gyel, rágja át magát a megfelelő info oldalakon, amelyek forrásban is megtalálhatók a CD-mellékleten.

Mielőtt azonban nagyon elemélyednének az m4 lehetőségeiben, figyelmeztetek mindenkit, hogy ez a program olyan, mint az ópium. Egy idő után az ember m4-függő lesz, és mindent ezzel akar megcsinálni, pedig az adott feladatra esetleg jobb eszközök is létezhetnek.

Aszalós László

Microsoft Script Control

COM(b)os kis hozzáférés

A különböző programnyelveken végzett fejlesztéseknél szinte minden esetben felmerül az igény, hogy egy alkalmazást — készüljön akár DOS, akár Windows alá — minél inkább interaktívvá tegyük. Ennek egyik formája, hogy a felhasználónak saját nyelvi formában közvetlen beavatkozási lehetőséget biztosítunk. Végső soron ezt oldják meg az interpreter segítségével működő programozási felületek is. Vannak azonban esetek, amikor nincs szükség egy teljes interpreterre, de néhány utasítás erejéig a felhasználó kezébe kívánjuk adni a vezérlés kulcsát, esetleg rövid, néhány utasításból álló programocskák megírását is lehetővé téve.

Arra, hogy a vezérlést hozzáférhetővé tegyük a felhasználók számára, számos megoldás van a számítástechnikában. Gondoljunk például a weblapokba beágyazott scriptekre, melyeket a böngészővel együtt telepített programrészek értelmezik, hogy különböző funkciókat lássanak el a böngésző használata során. Ha ezek mindenki által elérhetők lennének, akkor könnyebb lenne olyan programokat készíteni, amelyek képesek futtatni a böngészkőben értelmezett scripteket, bizonyos funkciókat pedig scriptek futtatásával lehetne megoldani. A legtöbb esetben azonban ezeknek a scriptértelmezőknek a böngészőtől vagy más gazdaprogramtól független hozzáférése meglehetősen körülményes.

Ezen segít egy külön erre a célra, a scriptek közvetítésére és ezzel futtatásuk megvalósítására kifejlesztett, COM (Component Object Model) alapú fejlesztői segédeszköz, a Microsoft Script Control. Ez teszi lehetővé olyan alkalmazások készítését, melyekben 32 biten a programozó elérheti az ActiveX alapú olyan scriptértelmezőket (script engine), mint az Internet Explorer által is használt, Visual Basic alapú VBScript, vagy a Java alapú JScript. De nem okoz problémát más nyelv alkalmazása sem, melyhez az említett, ActiveX alapú és a többi scriptértelmezővel azonos hívási konvenciókat alkalmazó segédlet rendelkezésre áll. Ilyen például a Perl elemeket használatba vevő PerlScript engine is. (Megtalálható az Új Alaplap 1998/3. számának CD-mellékletén.)

A Script Control segítségével azonban nem csupán a böngészőkből válik elérhetővé a scripthasználat, hanem minden olyan fejlesztőeszközben (például Visual Basic-kel) készült alkalmazás számára, amely képes a szabványos COM objektumokat kezelni. Erre a Script Control bináris állományával együtt települő help fájl is számos példát tartalmaz, teljes metóduskódokkal. Egy abból vett kódrészlet, amely egy gomb lenyomására MessageBox üzenetet ad, a következő:

```
Private Sub Command1_Click()
   Dim sc
   Dim strProgram As String
   strProgram = "Sub Main" & vbCrLf & _
      "MsgBox ""Hello World""" & vbCrLf & _
      "End Sub"
   Set sc = CreateObject("ScriptControl")
   sc.language = "VBScript"
```

```
    sc.addcode strProgram
    sc.run "Main"
End Sub
```

A kezdetben típus nélküli változóként deklarált "sc"-hez rendeljük hozzá a ScriptControl ActiveX vezérlőt, melynek később megadjuk az értelmezés nyelvét (alapértelmezése VBScript) és a végrehajtandó scriptutasítást ("sc.addcode"), majd futtassuk le a kódot ("sc.run"). Különösen figyelni kell arra, hogy a Script Control nem végez scripten belül szintaktikai ellenőrzést, ezért ha a megadott script hibákat tartalmaz, a futtatás sikertelen lesz. Erről a ScriptControl Error objektumából kaphatunk további információkat.

Arra is lehetőségünk van, hogy paraméterezett eljárásokat hajtassunk végre. Például az előző eljárást némileg módosítva lehetőség van egy Text mező (Text1) szövegének a megjelentetésére a MessageBoxban:

```
Private Sub Command1_Click()
    Dim sc
    Dim strProgram As String
    strProgram = "Sub Main (a)" & vbCrLf & _
        "MsgBox a,0,""Message from VBScript""" & _
        vbCrLf & "End Sub"
    Set sc = CreateObject("ScriptControl")
    sc.Language = "VBScript"
    sc.AddCode strProgram
    a = sc.Run("Main", Text1.Text)
End Sub
```

Ez a kódrészlet azt is jelzi, hogy bár a ScriptControl az adott program keretein belül fut, a script és a gazdaprogram tartalmazhat azonos nevű változót. Itt az "a = sc.Run" értékadásnak csak szintaktikai jelentősége van, bár az említett helpben a Visual Basic ilyen jellegű szintaktikai igényére nem találtam utalást. Ezt, hogy azonos környezetben futó ActiveX metódus visszaadja az értékét, felhasználhatjuk a scripten belüli folyamatokban is, amire példa a következő Perl scriptet futtató Visual Basic rutin:

```
Private Sub Command2_Click()
    Dim sc
    Dim strProgram As String
    Dim strReturn As String
    strProgram = "sub Main " & vbCrLf & _
        "{ " & vbCrLf & _
        " return shift;" & vbCrLf & _
        " }"
    Set sc = CreateObject("ScriptControl")
    sc.Language = "PerlScript"
    sc.AddCode strProgram
    strReturn = sc.Run("Main", "Ez pedig a PerlScript
üzenete")
    Text1.Text = strReturn
End Sub
```

A ScriptControl, akárcsak a Windows rendszerben regisztrált bármely más ActiveX komponens, más programnyelvekből is elérhető, tehát a Delphi 3-ból is. A palettára azonban nem sikerült felvennem, a kezdetben hibátlannak tűnő ActiveX konverzió eredményeként kapott *.Pas a fordításkor hibát jelezve leállt. Ennek talán az az oka, hogy a ScriptControl nem vizuális vezérlő. Használata azonban probléma nélkül megvalósítható volt a Delphi 3 CreateOleObject funkciójával. Ennek segítségével a Visual Basic példák Delphi-átirata a VBScript, illetve a PerlScript használatára:

```
procedure TForm1.Button1Click(Sender: TObject);
var
    sc : Variant;
    strProgram : WideString;
begin
    strProgram := 'Sub Main' +#13+
        'MsgBox "Hello World - VbScript" & _
        vbCrLf & vbCrLf & "from Delphi"'+
    ',vbOKOnly,"ScriptDemo"'+#13+
     'End Sub';
    sc := CreateOleObject('ScriptControl');
    sc.Language := 'VBScript';
    sc.AddCode (strProgram);
    sc.Run ('Main');
end;

procedure TForm1.Button2Click(Sender: TObject);
var
    sc : Variant;
    strProgram : WideString;
    strReturn : WideString;
begin
    strProgram :=
    'sub Main' + #13+
    '{ ' + #13 +
    ' return "Itt az üdvözlet a PerlScript -b&l"; '
    ' +#13+ }';
    sc := CreateOleObject('ScriptControl');
    sc.Language := 'PerlScript';
    sc.AddCode (strProgram);
    strReturn:=sc.Run ('Main');
    Edit1.Text := strReturn;
end;
```

Ezek az eljárások inkább csak ízelítők a scriptek használatából a különböző programokban. Bekérhetjük a scriptet a felhasználótól is (és akkor programunk csak közvetítő a végrehajtásban), vagy globális objektumként megadva a ScriptControlt, elérhetővé tehetjük programunk teljes futásidejére. Komplex eljárási rendszereket felépítve akár saját eljárásokkal is bővíthetjük a ScriptControl moduljait ("ScriptControl.Modules.Add"). Elhelyezhetünk komplett scripteket magában az EXE fájlban is, ahonnan azokat szövegként (Stringtable) bármikor kiolvashatjuk és lefuttathatjuk.

Végül álljon egy példa arra, hogy miképpen használhatjuk a leírtakat scriptek feldolgozására akkor, ha a scriptnyelv támogatja a COM kezelését. A Python esetében — ha telepítettük annak Win32Com és más Win32 kiegészítéseit — a Python parancssorában a következőket begépelve megjelenik a MessageBox:

```
>> import win32com.client
>> sc=win32com.client.Dispatch("ScriptControl")
>> strProgram='Sub Main \n MsgBox "VBScript
        a Pythonból",0,"Hello" \n End Sub'
>> sc.language='VBScript'
>> sc.addcode(strProgram)
>> sc.run('Main')
```

De mindezeket beépíthetjük egy önálló scriptfájlba is (például ScriptControl.py) a következő módon:

```
import win32com.client
#//////////////////////
#/ Demó az MS ScriptControl ActiveX használatára a
#/ Python nyelvből (c) Simay Endre István - Budapest
#//////////////////////
sc=win32com.client.Dispatch("ScriptControl")
strProgram = 'Sub Main \n MsgBox'+ ms+' \n End Sub'
sc.language = 'VBScript'
sc.addcode(strProgram)
sc.run('Main')
```

A 32 bites Windows platformon további scriptkezelési lehetőség is rendelkezésre áll. Ez a Windows rendszer használatát egyszerűsítő és programozhatóvá tevő Scripting Host, mely a rendszerbe integrált elem a Windows 98-ban, illetve az Internet Information Server 4.0-ban és a Windows 2000-ben. A Microsoft Windows Scripting Host szintén nyelvfüggetlen elérést biztosít. A nyelvfüggetlenség mögött ez esetben is egy ActiveX alapú objektum áll, amelynek telepítést követő regisztrálása teszi lehetővé a Scripting Host elérését a teljes Windows rendszer számára. A Scripting Host használatának segítésére a Microsofttól rendelkezésre áll egy teljes mintagyűjtemény is. Ez a dokumentáció mellett példaalkalmazásokat tartalmaz a Registry és a különböző Windows alkalmazások kezelésére egyaránt.

Simay Endre István

A Microsoft második nagy „ajándéka”

Windows scriptvírusok

Sokáig úgy tűnt, hogy az Internet vírusvédelmi szempontból viszonylag biztonságos, és a Weben való böngészés közben nem kell félnünk attól, hogy gépünket vírus fertőzheti meg. Most a Microsoft egy huszárvágással kinyitotta Pandora szelencéjét, rászabadította a világra az Interneten terjedni képes vírusok új családját. Már a makróvírusok is annak köszönhették létüket, hogy a Microsoft makrónyelvében könnyedén lehetett vírusokat vagy romboló rutinokkal rendelkező trójai programokat írni. Ha a fejlesztők egy kicsit is átgondolták volna az egészet, akkor szabályozottabb makróértelmező környezetet hoztak volna létre, amely nem teszi lehetővé a kártékony programok „szabadrablását”. Hogy a dolgok átgondolására nem teljesen képtelenek, azt bizonyítja az Office 97-hez kibocsátott Service Release 1, amely már valóban az értelmező szintjén szab korlátokat a makróvírusoknak. És lám, most megint egy nagy slendriánság...

Azt gondolhatná az ember, hogy a Microsoft nem fog még egyszer ugyanabba a csapdába beleesni, és amikor a HTML lapokba is beilleszthető VBScript nyelv specifikációját megalkotja, körültekintőbben fog eljárni. A VBScript nyelv (mint általában a Microsoft összes öngólja) eredetileg jó célokat volt hivatott szolgálni. A JavaScript nyelvhez hasonlóan HTML lapokba beszúrható kis programok nyelveként indult el, majd szervesen beillesztették az operációs rendszerbe, hogy annak (a legtöbb operációs rendszer script nyelvéhez és a régi DOS batch programjaihoz hasonlóan) legyen ütőképes programozói kiterjesztése. Mindezt Windows Scripting Host álnéven a Windows 98 és a Windows NT 5.0 már alapértelmezésben tartalmazza, a korábbi 32 bites rendszerek esetében pedig opcionális kiegészítőként letölthető és telepíthető.

Sajnos a Microsoft nem tanult az előzményekből: a VBScript nyelv és az interpreter is teljes hozzáférési lehetőséget kap a böngészőt futtató számítógép fájlrendszeréhez. Pedig csak egy kicsit kellett volna tanulmányozniuk a hasonló célokat szolgáló JavaScript megoldását. Ott ugyanis a teljes fájlkezelést kihagyták. Ez persze korlátokat szab a webprogramozók szárnyaló fantáziájának, ezért a Microsoft vélhetően úgy döntött, hogy a könnyebb programozhatóság (és ennek révén a webfejlesztők elcsábítása) érdekében nem szab ilyen korlátokat, kiterjeszti a programozási lehetőségeket. Hát nem is kellett sokáig várni arra, hogy ezt a védelmi lyukat kihasználó első vírusok megjelenjenek. Legfeljebb az a meglepő, hogy nem bukkantak fel még hamarabb. Erre egyetlen magyarázatot látok: az igazán komoly vírusírók méltóságon alulinak találták a scriptvírusokkal való foglalkozást („sas nem kapdos legyeket”), a kezdőknek pedig egyszerűen nem jutott eszükbe. Mindezt alátámasztja, hogy amikor az első példány megjelent (és ezzel mindenki számára világossá vált, hogy akár VBScriptben is lehet vírust írni), akkor két hét alatt egy tucat új scriptvírust gyártottak.

Csak összehasonlításképpen: az első makróvírus, a Concept megjelenése után három hónappal még csak 3 makróvírus létezett, és körülbelül fél év kellett ahhoz, hogy számuk a tucatot elérje. A Windows vírusok esetében majdnem egy év telt el az első pédány megjelenésétől a tucat eléréséig. És most aligha lehet arra számítani, hogy a lelkesedés elapad, mert az Interneten keresztül történő fertőzés a kezdő vírusírók fantáziáját megmozgatja, és az eddig megjelent példányok alapján egy kutyaütő is össze tud hozni 5 perc alatt egy új változatot. Ezt még jobban megkönnyíti, hogy ezek a vírusok szabadon módosítható forráskód formájában terjednek. A VBScriptvírusok akár a makróvírusoknál is nagyobb karriert futhatnak be, pedig ez utóbbiakból a gyenge első fél év után 3 év alatt több mint 3000-et készítettek. Elég sötét jövőkép tárul fel előttünk, és ezt nem a vírusszakértők kötelező pesszimizmusa mondatja, hanem a vírusírás tendenciáiból leszűrhető tapasztalat.

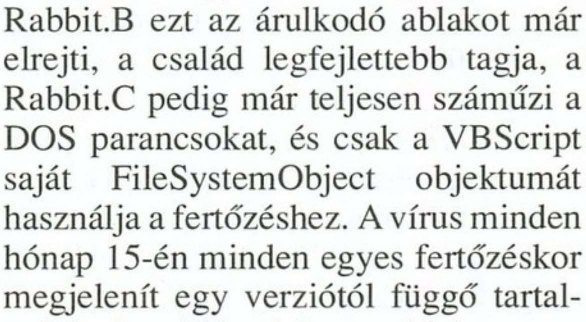
A Rabbit család

Időrendben a VBScriptvírusok az első példányai a Rabbit család tagjai voltak: Rabbit.A, Rabbit.B és Rabbit.C (vagy ahogy szerzőjük eredetileg elnevezte őket: VBSV 1., 1.1 és 2.0). A család tagjai az aktuális könyvtár *.VBS állományait megtámadó, egyszerű felülíró vírusok, azaz a kiszemelt célpontok eredeti tartalmát megsemmisítik, felülírják.

A Rabbit.A változat egyszerűen csak egy DOS task létrehozásával, és abban egy batch parancs kiadásával oldja meg terjedését, emiatt minden fertőzéskor egy (a lelepleződést megkönnyítő) DOS ablak ugrik fel a képernyőre. A Rabbit.B ezt az árulkodó ablakot már elrejti, a család legfejlettebb tagja, a Rabbit.C pedig már teljesen száműzi a DOS parancsokat, és csak a VBScript saját FileSystemObject objektumát használja a fertőzéshez. A vírus minden hónap 15-én minden egyes fertőzéskor megjelenít egy verziótól függő tartalmú, itt lent látható üzenetablakot.

Az ablak megjelenítése után a Desktopon létrejön egy shortcut, amely a vírusíró által fémjelzett CodeBreakers vírusíró csoport weblapjára mutat. Sőt a B és C variáns csatlakozik is erre a címre, megjelenítve az alapértelmezett böngészővel.

A vírusnak érdekes mellékhatása van. A böngészőprogramok ugyanis a cache-ükben levő fájlokat úgy tekintik, mintha azok az aktuális könyvtárban lennének. A vírus emiatt a cache-ben lévő összes .VBS fájlt megfertőzi, és az aktuális könyvtárba írja vissza azo-



kat. Ez a böngészők esetében úgy tűnik, hogy a Desktop. Emiatt ha valaki sok .VBS fájlt nézeget a böngészőjével (és megtelik velük a cache), akkor fertőzés esetén a vírus egy szemvillanás alatt telepakolja a Desktopot a fertőzött VBScript programokkal.

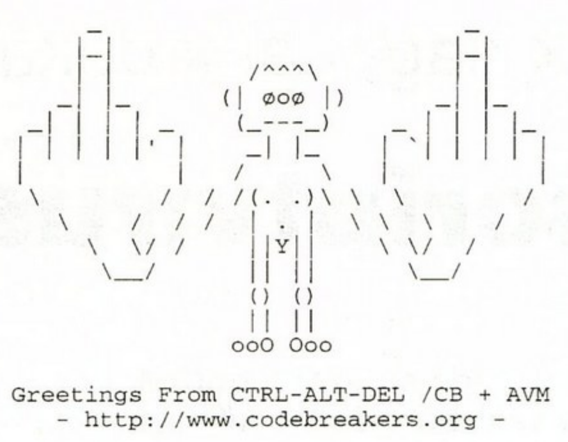
VBSv777 vírus

Nem sokkal később, egy másik vírusíró jóvoltából megjelent a VBSv777 vírus, amely a Rabbit családnál már jóval komolyabb ellenfél. A 3914 bájt effektív hosszúságú vírus a *.VBS állományok elejére szúrja be magát. Ez az interpreteres környezet miatt szükséges, ugyanis egy script futása a fájl legelején kezdődik, az viszont nem könnyen határozható meg, hogy hol ér véget. A nem felülíró script vírusok túlnyomó többsége ezért várhatóan az eredeti programok elejére fogja beírni magát, ezzel némi könnyebbséget nyújtva a víruskeresőknek.

Aktivizálódása után a VBSv777 célba vesz minden *.VBS állományt az aktuális könyvtárban, a Windows könyvtárában, valamint az itt található Profiles\All Users\Desktop, Profiles\Administrator\Desktop, illetve Desktop könyvtárakban (a két előbbi tartalmazza Windows NT alatt a rendszergazda által definiált Desktopot, illetve az összes felhasználó számára érvényes Desktopot, az utolsó pedig a Windows95/98 alatt definiált Desktopot). Jórészt ezekben a könyvtárakban helyezi el a Windows Scripting Host a felhasználó életét könnyíteni hivatott programocskákat.

A vírus minden egyes potenciális áldozatfájl esetében ellenőrzi, hogy annak legelső 8 bájtja megegyezik-e a víruséval, ami a 'VBSv777 string. Ha ezt nem találja, akkor még érintetlen a script, így nekilát a fertőzésnek. Először beolvassa az eredeti script teljes szövegét, majd felülírja azt a lemezen a víruskóddal, ezután pedig a fájl végére biggyeszti az eredeti programot.

Ez a vírus már kifejezetten romboló hatású. Minden hónap másodikán úgy lép működésbe, hogy ha egy fertőzés 9 órakor történik, megjeleníti a mellékelt üzenetablakot, majd az összes lokális merevlemezen található *.TXT és *.DOC állomány végére beilleszti a



fenti, a szerző által vélhetően humorosnak szánt grafikát. Valódi szövegfájloknál ez az aktus bosszantó ugyan, de ártalmatlan, a Word dokumentumokat viszont annyira megbolygatja, hogy azok használhatatlanokká válnak, amit a károsultak aligha tartanak jó tréfának.

Happy vírus

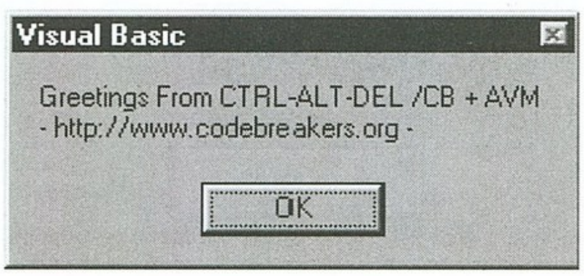
Amint előzőleg már szó volt róla, a scriptvírusok többsége feltehetően a célpontok elejére szúrja be magát. A kevés kivétel egyike a harmadikként felbukkant vírus, a Happy volt, amely a scriptek végére illeszti kódját. Ez a vírus szempontjából nem a legsikeresebb megoldás, mert ha az eredeti script több eljárásból áll, akkor semmi nem garantálja, hogy a fájl végén csücsülő vírusra átkerül a vezérlés. De már megszokhattuk, hogy a vírusírók sem mindig foglalkoznak ilyen részletkérdésekkel, meg ha csak minden harmadik példány lesz is működőképes, az már garantálja a fertőzés terjedését.

A Happy szinte minden szóba jöhető könyvtárat megcéloz. Közéjük tartozik az aktuális meghajtó gyökérkönyvtára, a Desktop, a MyDocuments, a Startup, a scriptet tartalmazó könyvtár, valamint a PATH környezeti változóban definiált valamennyi könyvtár. A Happy a víruskód végére biggyesztett A karaktert használja a célpontok fertőzöttségének ellenőrzésére. Mivel a scriptek végére írja magát, nyilván ott kell keresnie az azonosítót is. Ha nem találja, akkor fertőzetlennek tekinti a scriptet, és a végére bemásolja a víruskódot. Ehhez azonban trükkökre van szüksége. Ugyanis a hagyományos vírusokkal ellentétben az ilyen vírus nem férhet hozzá a vírusnak a memóriában lévő képéhez, hanem a futtatott fertőzött scriptnek a merevlemezen lévő példányából kell azt kiolvasnia. De a vírus a script végén van, ezért meg kell határoznia, hogy hol kezdődik a vírus. Ezt a víruskód elejére biggyesztett '§ azonosító jelzi. A vírus mindezek alapján a fertőzendő script végére tudja illeszteni a kódját. Dolga végeztével még ellenőrzi, hogy létezik-e egy esetleges korábbi fertőzésből származó, pusztán a víruskódot tartalmazó AVM.VBS fájl. Ha a gép még szűz, vagy ha valaki időközben letörölte ezt a fájlt, a vírus létrehozza, utána belemásolja a víruskódot, majd a regisztrációs adatbázisban a HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\AVM kulcs alatt bejegyzi, hogy a vírust tartalmazó (és az imént láthatóan garantáltan létező) AVM.VBS script a Windows minden indításakor lefusson. A vírus egyébként a terjedésen kívül semmi mást nem csinál.

A fenti 3 alapvírus megjelenése után rövidesen több hasonló is felbukkant. A HTMLOffline egy VBScript kiegészítéssel ellátott HTML vírus. Ha a weblapot a böngészőben megnyitjuk, az aktuális könyvtárban levő összes *.HTM és *.HTML állományt felülírja a víruskóddal. Az 1nfect vírus a Windows driverek telepítését definiáló *.INF fájlokat fertőzi meg. A telepítés mechanizmusát figyelembe véve ezek a speciális fájlok tartalmazhatnak a Windows telepítőprogram által értelmezett olyan parancsokat, amelyekkel az AUTOEXEC.BAT állomány tartalmát lehet megváltoztatni. Nos az 1nfect ezt úgy végzi el, hogy a végére biggyeszt néhány utasítást, melyek révén a következő rendszerindításkor a Windows\Inf könyvtárban lévő legújabb információs állományt felülírja a vírus kódjával.

Hála a Microsoft által erőteljesen propagált, és minden elérhető helyre (sajnos a VBScriptbe is) besuvasztott ActiveX technológiának, egy scriptből gyakorlatilag minden ActiveX alkalmazás vezérelhető. Mivel a Word 97 is ilyen, logikus lépés volt a vírusírók részéről, hogy olyan kétéltű vírusokat próbáljanak írni, amelyek a VBS fájlok mellett Word dokumentumokat is megfertőznek. Az első próbálkozás egy makróvírus dropper, az Insert. Ez pusztán annyit csinál, hogy a fertőzött weblapon lévő program futtatásakor létrehoz egy dokumentumot, és abba beültet egy, a Class osztályba tartozó, nagyon primitív makróvírust. Ez egyetlen Document_Close nevű makróból áll, amely a dokumentumok bezárásakor fertőz. A továbbiakban sem az Insert, sem a létrehozott makróvírus nem fog VBS fájlokat fertőzni.

Lényegesen bonyolultabb a ConceptII fedőnévre hallgató változat, amely már igazi átjárást biztosít a VBS scriptek és a DOC makróvírusok között. Az ezt tartalmazó HTML lap megnyitása után megfertőzi az aktuális könyv-



tárban található valamennyi .VBS állományt, majd a Wordöt veszi célba. Miután ellenőrizte annak fertőzetlenségét (erre a célra a víruskód elején található 1nternal stringet használja), a NORMAL.DOT ThisDocument moduljába másolja magát. A vírus két részből áll, a VBScript és a Document_Close makrót tartalmazó VBA komponensből. Ezek közvetlenül egymás után találhatók a forráskódban (ne felejtsük el, hogy ezek a vírusok forráskód formájában terjednek), ezért mindkét komponens bekerül a modulba. A VBSript és a VBA közötti minimális különbségek miatt meg kell oldania, hogy egyszerre mindig csak a megfelelő rész futhasson le. Ennek érdekében a bemásolás után a VBScript fázishoz tartozó sorok elé REM-eket illesztve kizárja azokat a végrehajtásból. A Word fázisban a ConceptII jóravaló makróvírusként teszi a dolgát. Bezáráskor megfertőz minden dokumentumot (ennek során is a kód legelején található 1nternal szavacskát tekinti a fertőzöttség jelének). Ha ez 15-én történik, akkor a vírus átlép a VBScript fázisba: végigkeresi a C: meghajtó valamennyi .VBS állományát, és azok tartalmát felülírja a vírus kódjával.

A Microsoft nem először döntött úgy, hogy a gyorsabb piaci megjelenés, a kényelem és a könnyebb kezelhetőség érdekében sutba dobja a biztonsági megfontolásokat. És nem először reagál úgy a következményekre, hogy Pilátusként mossa kezeit, a vírusvédelmi rendszerek fejlesztőitől pedig elvárja, hogy a Microsoft hibájából keletkezett biztonsági réseket helyette befoltozzák.

Szappanos Gábor

Az élen a helyzet változatlan

Víruskeresők mustrája

A hamburgi Virus Test Center ismét közzétette rendszeres víruskereső tesztjei közül az 1998. októberinek a végeredményét. Felsorolom azt is, hogy ehhez hány vírust használtak fel, mert abból rögtön érthető lesz, hogy miért tart ilyen sokáig a tesztek kiértékelése.

A hamburgi Virus Test Center nyers vírusismereti tesztjéhez felhasznált források:

— 13 993 programvírus, összesen 112 038 fertőzött programban,
— 881 bootvírus,
— 2159 makróvírus,
— 3300 tiszta program a téves riasztások felmérésére.

Mivel a vírusok túlnyomó többsége laboratóriumi példány, és csak tíz százalékuk fordul elő „a vadonban", ezért a tényleges fertőzéseket okozó vírusokra vonatkozóan külön is ellenőrizték a víruskeresők tudását. Az ennek során felhasznált vírusok és konrollprogramok:

— 122 programvírus,
— 207 bootvírus,
— 75 makróvírus,
— 325 tiszta makróprogram a téves riasztások felmérésére.

Sok fejlesztő szerint eleve csak „az életből vett" vírusokat kellene vizsgálni, hiszen azok reálisabban mutatják, hogy a felhasználókat fenyegető veszélyekkel hogyan birkóznak meg az egyes antivírus programok. Az összes laboratóriumi vírus detektálása szerintük felesleges, mert azok többségével a vírusgyűjtőkön kívül úgysem fog soha senki szembetalálkozni.

Az alábbi táblázat ismerteti, hogy a legismertebb, és hazánkban is hozzáférhető víruskeresők a teljes víruskészlet hány százalékát ismerték fel, külön részletezést adva a programvírusokra és a makróvírusokra. A táblázatban szerepel az utóbbi négy teszt eredménye is, valamint a legutolsóhoz képest tapasztalt változás.

Amint látható, abszolút első a teszt időpontjában még önállóan létező, de azóta bekebelezett Dr.Solomon's AVTK. Nyilvánvaló, hogy a Network Associates jó vásárt csinált, hiszen az is kiderül a táblázatból, hogy már szüksége volt erre a vérfrissítésre. A második a hagyományosan jól szereplő Antivirus Pro, de nincs okuk szégyenkezni a Norton Antivirus és a Sweep fejlesztőinek sem.

Szappanos Gábor

Kereső	Programvírusok					Makróvírusok				
	97/02	97/07	98/02	98/10	Változás	97/02	97/07	98/02	98/10	Változás
Antivirus Pro	98,5	98,4	99,3	99,7	+0,4	99,3	99,0	99,9	100,0	+0,1
DrWeb	93,2	93,8	92,8	93,1	+0,3	90,2	98,1	94,3	99,3	+5,0
Dr.Solomon's	99,7	99,6	99,9	99,9	0,0	97,9	98,9	100,0	100,0	+0,0
F-Prot	90,7	89,0	96,0	95,5	–0,5	43,4	36,1	99,9	99,8	–0,1
F-Secure	–	–	99,4	99,7	–	–	–	99,9	90,1	–9,8
InocuLAN	–	–	92,0	93,5	+1,5	–	–	90,3	95,2	+4,9
Norton Antivirus	66,9	67,1	97,1	98,1	+1,0	80,7	86,4	98,7	99,8	+1,1
Norman Virus Control	87,4	89,7	94,1	93,8	–0,3	13,3	96,6	99,2	90,8	–8,4
Scan	83,9	93,5	90,7	87,8	–0,9	95,1	97,6	99,0	98,6	–1,4
Sweep	95,9	94,5	96,8	98,4	+1,6	87,4	89,1	98,4	98,6	–0,2
TBAV	95,5	93,7	92,1	93,2	+1,1	72,0	96,1	99,5	98,7	–1,2

Excel — felsőfokon IX.

Hab a tortán

Ebben a befejező részben néhány olyan programozási fogásról lesz szó, amelyek nélkül meg lehet ugyan élni, alkalmazásukkal viszont kényelmesebbé és biztonságosabbá tehetjük programjaink használatát. Megnézzük, hogyan tudunk programjainkhoz illeszteni egyéni eszköztárat, illetve milyen eszközökkel tudjuk automatizálni futásukat. Végül pedig néhány profi mentési technikáról esik szó.

Testre szabott eszköztár létrehozása

Elegáns megoldás, ha lehetővé tesszük a felhasználónak, hogy gyakran használt programjait egy-egy ikonnal indíthassa; evégett saját ikonokat kell létrehoznunk. Feltételezem, hogy az olvasó ismeri az eszköztárak, és ezen belül az ikonok kézi átalakításának módszereit; ezekről különben az 5. részben röviden már volt szó. Ezúttal természetesen programmal oldjuk meg a feladatot. Legegyszerűbben akkor járunk el, ha egy eljárással kialakítunk egy saját (egyéni) eszköztárat, és ezt a rutint elhelyezzük az aktuális munkafüzetben; célszerűen külön modullapon, hogy könnyen lehessen módosítani. A rutint úgy írjuk meg, hogy a munkafüzet megnyitásakor *automatikusan* (lásd alább) feltegye a szükséges ikonokat az eszköztárra.

Illik tudni, hogy az utoljára használt eszköztárat az Excel megőrzi az EXCEL5.XLB állományban; ezért a tisztességhez hozzátartozik, hogy egyéni alkotásunkat a munka befejeztével eltávolítsuk, és visszaállítsuk az eredeti (rendszerint a „Szokásos" és a „Formázó" eszköztárat tartalmazó) alapállapotot.

Új eszköztárat létrehozni, és a lap tetején, a harmadik sor bal szélén megjeleníteni az alábbi parancsokkal lehet:

```
Toolbars.Add "eszköztárnév"
With Toolbars("eszköztárnév")
   .Position = xlTop
   ' Lehet még: xlBottom, xlRight, xlLeft
   .Top = 53
   ' Az első sor y koordinátája 13, a másodiké 33
   .Left = 1    ' Az első pozíció x koordinátája
   .Visible = True   ' Így lesz látható
End With
```

Újdonsült eszköztárunkba egyenként kell „becsalogatnunk" a használni kívánt ikonokat, kódszámuk megadásával; egyidejűleg engedélyezhetjük vagy letilthatjuk használatukat, és azt is beállíthatjuk, hogy induláskor az „ikongomb" lenyomott vagy felengedett állapotban legyen-e. Az alábbi utasításokkal a „lámpa" és a „jegyzettömb" ikont helyezzük el felengedett állapotban (`Pushed` kulcsszó) a már létező eszköztár első két pozíciójában (`Before` kulcsszó), és engedélyezzük a használatukat (`Enabled` kulcsszó) úgy, hogy lenyomásukra (`OnAction` kulcsszó) elindul egy-egy makróprogram:

```
With Toolbars("eszköztárnév")
  With .ToolbarButtons
    .Add Button:=179, Before:=1, _
      OnAction:="prog1név", _
      Pushed = False, Enabled = True
    .Add Button:=217, Before:=2, _
      OnAction:="prog2név", __
      Pushed = False, Enabled = True
  End With
End With
```

A felengedett állapot és a használat engedélyezése alapértelmezés, nem kell külön megadni. A Súgó „Ikonkategóriák jegyzéke" fejezetében megtalálható a beépített ikonok kódszáma, kivéve az „egyéni" csoportot; ezekét a 12. táblázatban adom meg. Természetesen módosíthatjuk a beépített eszköztárakat is, nevüket Excel konstanssal lehet megadni (`xlStandard`, `xlFormat` stb.), de hivatkozhatunk rájuk egyszerűen a sorszámukkal, amit a <u>*Nézet*</u> - *Eszköztárak* menüpont választására megjelenő ablakból állapíthatunk meg.

Az ikonokat `Copy` vagy `Move` metódussal mozgathatjuk egyik helyről a másikra, akár különböző eszköztárak között is. Az alábbi utasítással például a „Standard" eszköztár 4. ikonját átvisszük saját eszköztárunk végére:

```
Toolbars(xlStandard).ToolbarButtons(4)_
  .Move Toolbars("eszköztárnév"),
Toolbars("eszköztárnév").ToolbarButtons_
  .Count + 1
```

Az ikonképet a `CopyFace` metódussal lehet a vágólapra tenni, ahonnan a `PasteFace` metódussal másolhatjuk rá egy másik ikonra.

12. táblázat: Az egyéni eszköztár ikonjai

Magyar neve	Angol neve	Kódszáma
Boríték	Envelope	230
Bögre	Mug	218
Csengő	Bell	226
Hal	Fish	214
Hangszóró	Speaker	209
Injekciós tű	Tack	215
Integrál	Integral	200
Jegyzettömb	Notebook	217
Kamera	Camera	228
Káró	Diamond	222
Kéz	Hand	229
Kőr	Heart	221
Léggömb	Balloon	219
Mentés	Save	207
Mosolygó	Smile	211
Nyitás	Open	206
Óra	Clock	213
Paletta	Palette	97
Pasziánsz	Solitaire	201
Pikk	Spade	223
Számológép	Calculator	237
Szemetes	Trash	225
Szomorú	Frown	212
Telefon	Telephone	220
Treff	Club	224
Üres	Blank	231
Visszavonás	Undo	208

Egyéni eszköztárat vagy annak mondjuk egy ikonját törölni az alábbi utasítással lehet:

```
Toolbars("eszköztárnév").Delete
Toolbars("eszköztárnév").ToolbarButtons_
  (ikonsorszám).Delete
```

Ha jól felépített vagy egy-egy feladathoz illesztett eszköztárunkat meg akarjuk őrizni, a munkafüzet bezárása után másoljuk át az EXCEL5.XLB állományt más néven, egy célszerűen szintén .xlb kiterjesztésű állományba. Ha ismét saját eszköztárunkat akarjuk használni, egyszerűen csak nyissuk meg ezt a másik állományt.

Automatizálási technikák

Munkafüzet automatikus megnyitása

Az Excel megnyitásakor minden munkafüzetet megnyit, amelyik az ...\EXCEL\XLSTART indítókönyvtárban van; tehát egyszerűen ide kell másolni az automatikusan elindítandó állományokat. Mivel ezek az állományok olyankor is megnyílnak, amikor más munkát akarunk végezni, ez a módszer sokszor inkább bosszantó, mint előnyös. Célszerűbben járunk el, ha a gyakran használt munkafüzeteinket egy-egy ikonhoz rendeljük, és az ikon(oka)t elhelyezzük abban a Windows-csoportban, amelyben az Excel indítóikonja is van. (Ha egy munkafüzet nem az Excel főkönyvtárában van, elérési útvonalát is meg kell adni.) A beépített ikonokat a Windows Rendszer csoportjában levő <u>*File*</u> - *Új* - *Program* menüpont választására előugró „Programjellemzők" ablak IKONOK... nyomógombjával lehet előcsalogatni; akinek ez a választék nincs ínyére, az említett ikonszerkesztési eljárásokkal tetszőleges ikonképet alakíthat ki magának. Ha egy ilyen ikonra kattintunk (bal egérgomb), automatikusan elindul az Excel, és megnyitja a kívánt munkafüzete(ke)t.

Automatikus nyitási és zárási eljárások

Egy felhasználóbarát program indításakor automatikusan megteszi a szükséges előkészítő lépéseket (például beállítja saját eszköztárát, megnyitja a működéséhez szükséges aktualizált adatállomány(oka)t stb.), és rögtön el is indítja a főmenüt megjelenítő főprogramot; befejeződésekor pedig automatikusan elvégzi a szükséges mentéseket és visszaállításokat. Ezeket a tevékenységeket az (automatikusan) megnyitott munkafüzetben elhelyezett, kötött nevű

Sub Auto_Open ( ), illetve ***Sub Auto_Close ( )***

eljárással lehet végrehajtani; az előbbi a munkafüzet kézi megnyitásakor, az utóbbi kézi lezárásakor indul el automatikusan. (Az eljárásokat természetesen magunknak kell összeállítanunk, a tanult szabályok szerint.)

Az UAXLMIP6.XLS – *INDÍTÓ* munkalapján látható ***Auto_Open*** rutin a munkafüzet megnyitásakor 4 ikonból álló saját eszköztárat készít és jelenít meg, az ***Auto_Close*** rutin pedig visszaállítja az eredeti eszköztárakat.

Fontos tudnunk, hogy ezek az automatikus eljárások nem indulnak el, ha a munkafüzetet programból nyitjuk meg, illetve zárjuk le. Ebben az esetben a megnyitó, illetve a lezáró munkafüzetben kell lennie egy eljárásnak, amely az „*mfnév*" nevű munkafüzetben elindítja a megfelelő rutint, a következő utasítással:

```
Workbooks("mfnév").RunAutoMacros [which
:=] {Auto_Open Auto_Close}
```

Két további megjegyzés:

— Egy munkafüzetben legfeljebb egy ***Auto_Open*** és egy ***Auto_Close*** rutin lehet; ha valamelyikből egynél több van, egyik sem fog lefutni.

— Ha (például tesztelés során) nem akarjuk, hogy lefusson az ***Auto_Open*** rutin, a munkafüzet megnyitásakor tartsuk lenyomva a SHIFT billentyűt.

Eseménykezelő eljárások

Sokkal hatásosabbak és rugalmasabban használhatók azok a metódusok és sajátságok, amelyek valamilyen esemény bekövetkeztekor — például egy időpont elérésekor, egy munkalap megnyitásakor, egy munkafüzet kiértékelésekor stb. — automatikusan elindítanak, aktiválnak egy *eseménykezelő* eljárást. Az aktiváló metódusok és sajátságok neve mindig az „on" (≈ hatására) szócskával kezdődik.

Ilyen eljárásokkal már találkoztunk, amikor egy munkafüzetben elhelyezett nyomógombhoz (például az UAXLMIP6.XLS – *FORGALOM* munkalapon) az `OnAction` sajátsággal programból rendeltünk hozzá egy eseménykezelő rutint; az aktiváló esemény az, hogy lenyomjuk a gombot. (Hasonlóan működik az `On Error` utasítás is, amely egy hibaesemény hatására indít el egy programrészletet.)

Most azokat a *szervező eljárás*okat tekintjük át, amelyek segítségével automatizálni lehet egy eseménykezelő rutin elindítását. Mivel érvénybe léptetésükhöz egyszer le kell futniuk, következik, hogy ha automatikusan akarjuk használni, az ***Auto_Open*** eljárásba kell beépítenünk őket; ezt a továbbiakban nem említem. Ha a szervező eljárás lefutott, a hozzá kapcsolt eseménykezelő eljárás mindannyiszor automatikusan le fog futni, valahányszor az esemény bekövetkezik, ezt a folyamatot *eseménykövetés*nek nevezzük. Ha le akarjuk állítani az eseménykövetést, ugyanazt a szervező eljárást kell használnunk, amellyel elindítottuk. Az alábbiakban egy-egy mintapéldát adunk a szervező eljárások használatára. (Az eseménykezelő eljárásokat csak jelképesen jelezzük, mivel tartalmuk a konkrét alkalmazástól függ.)

Automatikus időzítés: OnTime metódus

Az ***időzítő*** nevű szervező eljárás minden nap reggel 8 órakor elindít egy ***bejelentkezés*** nevű — itt pontosan meg nem határozott tartalmú — eseménykezelő eljárást:

```
Sub időzítés ( )
  Application.OnTime _
  EarliestTime := TimeValue ("8:00:00"), _
  Procedure := "bejelentkezés", _
  LatestTime := TimeValue ("8:15:00")
End Sub
...
Sub bejelentkezés ( )
  regisztrálási, munkaelőkészítési stb. _
  tevékenységek
End Sub
```

Az eseménykezelő rutin lefutásához az Excelt előzőleg természetesen el kell indítani. Mivel a szervező hatása csak az Excel lezárásáig tart, az ***Auto_Open*** eljárást az indítókönyvtárban levő munkafüzetben kell elhelyezni, így minden indításkor automatikusan lefut.

Ha az eseményként megjelölt időpontban éppen más eljárás fut, az Excel a `LatestTime` paraméterben megadott időpontig várakozik az eseménykezelő elindítására; ezt követően már nem hajtja végre. Ha viszont nem adunk meg ilyen paramétert, az eseménykezelő mindenképpen lefut, amikor az Excel legközelebb felszabadul.

Egyidejűleg több időpontra szóló időzítés lehet érvényben. Ha a szervező és az eseménykövető eljárás nem ugyanabban a munkafüzetben van, az útvonalat is meg kell adni.

Az elindított időzítés egy ilyen programmal leállítható:

```
Sub stopidőzítés ( )
  Application.OnTime EarliestTime :=
TimeValue ("8:00:00"), _
    Procedure := "bejelentkezés", _
    Schedule := False
End Sub
```

Munkalapváltás követése: OnSheetActivate és OnSheet-DeActivate tulajdonság

A ***váltásfigyelés*** szervező lefutása után a ***lapváltás*** eljárás mindig lefut, ha a TESZT.XLS nevű munkafüzetben a felhasználó aktivál egy munkalapot, vagy átvált egyik lapról a másikra; de a ***mostaztán*** eljárás csak akkor fut le, amikor az *EZAZIGAZI* nevű lap lesz aktív:

```
Sub váltásfigyelés ( )
  Workbooks("teszt.xls").OnSheetActivate _
  = "lapváltás"
  End Sub
...
Sub lapváltás ( )
  If  ActiveSheet.Name = "ezazigazi"
  Then mostaztán
  End If
End Sub
```

Az UAXLMIP6.XLS - *INDÍTÓ* munkalapon az `OnSheet-Activate` utasítás az ***Auto_Open*** eljárásban van; hatására mindig elindul a ***lapváltás*** nevű eseménykezelő eljárás, ha a munkafüzetben lapot váltunk. Az eljárás viszont csak akkor jeleníti meg a saját („uaxlmip" nevű) eszköztárat, ha az aktivált lap indexe 5-nél nagyobb, vagyis ha a *FORGALOM, LISTA, VALUTA* vagy *KONVERTA* lapot aktiváltuk, mivel ezeken vannak azok a rutinok, amelyeket az egyes ikonokkal elindíthatunk. A többi lap aktiválásakor az eszköztár eltűnik.

Az `OnSheetDeActivate` tulajdonság hasonlóan működik, csak akkor fut le, ha a felhasználó elhagy egy munkalapot.

Ügyeljünk rá, hogy munkafüzetváltáskor az új munkafüzet aktív lesz, mielőtt az előző munkafüzetben lefutna az `On-SheetDeActivate` tulajdonsággal elindított eljárás; így az ebben esetleg előforduló `ActiveSheet` és `Active-Workbook` parancsot az Excel hibásan értelmezheti!

„Gyorsbillentyű" használata: OnKey metódus

A ***specbill*** eljárás mindannyiszor elindítja az ***F12billen*** rutint, valahányszor a felhasználó lenyomja az F12 billentyűt:

```
Sub specbill ( )
  Application.OnKey  [Key :=] "{F12}", _
  [Procedure :=] "F12billen"
End Sub
...
Sub F12billen( )
  MsgBox "Az F12 billentyűt lenyomták!", _
  vbIformation
End Sub
```

Ha a sorrendet betartjuk, a `Key` és a `Procedure` paraméternév elhagyható. A `Key` paraméterben idézőjelek között kell megadni a látható karaktereket; a speciális karakterekre és a nem láthatókra pedig kapcsos zárójelbe írt szimbolikus nevükkel kell hivatkozni. A szimbolikus név általában megegyezik a billentyűre írt angol nevével, például INSERT, ESC, DEL, PGUP, F1 - F16 stb. A kivételeket és a kombinációs jeleket a 13. táblázat tartalmazza.

Az előírt billentyű(k) leütésére az `OnKey` eseménykezelő nem fut le, ha másik eljárás van folyamatban!

Egy billentyű normál értelmezésének visszaállításához a `Procedure` paraméter helyén üres füzért ("") kell megadni.

Adatmegadás figyelése: OnEntry sajátság

Ezzel a sajátsággal egy munkalap adatainak megváltozását követhetjük nyomon. Az UAFELAD6.XLS – *EGYMODUL* (rejtett) modullapján levő ***Auto_Open*** eljárásban elhelyezett

```
ThisWorkbook.Sheets("rendez").OnEntry = _
 "mostrendez"
```

13. táblázat: Gyorsbillentyű-szimbólumok

Speciális karakter	Szimbolikus jelölés
TABULÁTOR	"{TAB}"
ENTER (szürke)	"{ENTER}"
ENTER	"{~}"
BACKSPACE	"{BS}"
Le, fel, jobbra, balra nyíl	"{DOWN}" "{UP}" "{RIGHT}" "{LEFT}"
SHIFT és jobbra nyíl együtt	"+{RIGHT}"
CONTROL és pluszjel együtt	"^{+}"
CTRL és ALT és DEL együtt	"^%{DEL}"
ALT a kapcsos nyitó zárójellel	"%{{}"
Idézőjel	"{"}"

utasítás hatására mindannyiszor elindul a ***mostrendez*** eljárás, valahányszor a *RENDEZ* munkalapon változás történik. Az eljárásnak azonban csak akkor van látható hatása, ha a (sárgával jelölt) A3:A5 tartomány egyik cellájának értékét változatjuk meg: ekkor ugyanis növekvő sorrendbe rendezi a (szürke alapszínű) D3:D9 tartományban levő adatokat; ezek — egy kivételével (= a mai dátum napértéke) — valamilyen módon függenek az A3:A5 tartomány celláinak értékétől. (A függvények neve az E3:E9 tartományban látható.)

Ezzel kapcsolatban emlékeztetek rá, hogy az Excel automatikusan kiértékeli a munkalapon használt összes beépített függvényt, ha a munkalapon, valamint a `Volatile` típusú saját függvényeket (lásd 5. rész), ha a munkafüzetben megváltoztatunk egy adatot. Ennek illusztrálására szolgál a ***számláló*** függvény (ugyancsak az *EGYMODUL* modullapon), amely megszámlálja, hogy hányszor változott meg a *RENDEZ* munkalapon az A3 - A5 cella tartalma; az eredmény az E11 – G11 cellában látható. Ez a függvény azért is említést érdemel, mert példa arra, hogyan lehet egy **`Variant`** típusú függvénnyel tömböt visszaadni: a tömb elemei a hívást tartalmazó cellától jobbra eső cellákba kerülnek, sorfolytonosan. (Bemenő paraméterként adott egydimenziós tömb oszlopfolytonos is lehet; a kétdimenziós tömb természetszerűen téglalap alakú tartomány.) A hívó képleteket a tömbfeltöltés szabályai szerint kell feltölteni, ahogyan az UAFELAD0.XLS – *TÖMBÖK* munkalapján adott példák mutatják. (Jelöljük ki az E11:G11 tartományt, az E11 cellába írjuk be a függvényhívó "= számláló( )" képletet, és nyomjuk le egyszerre a SHIFT - CTRL - ENTER billentyűt.)

Amiről még szólni kell

Védjük, ami védhető

Ha egy munkalapon olyan információkat tárolunk, amelyeket nem kívánunk másokkal megosztani, rejtsük el a megfelelő sor(oka)t vagy oszlop(oka)t a *Formátum – Sor/ Oszlop – Elrejtés* almenü választásával; képleteink eltüntetésére használjuk a *Formátum – Cellák* menü „Cellák formázása" ablakának „Védelem" lapján a „rejtett" jelölést; az objektumok láthatóságát pedig az *Eszközök – Egyebek* menü „Látvány" ablakának „Objektumok" mezőjében levő választógombokkal szabályozhatjuk („látszanak" – „helyük látszik" – „rejtve maradnak"); de egész lapokat is elrejthetünk (*Formátum – Lap – Elrejtés*, illetve modullapok esetében *Szerkesztés – Lap – Elrejtés* almenü).

Mindezen fáradozásunk persze teljesen hiábavaló, ha olyan programozóval akadunk össze, aki olvasta ezt a cikksorozatot, mert ő tudni fogja, hogy mindegyik felsorolt esethez tartozik „Felfedés" almenüpont is, amivel minden rejtett elemet ismét

láthatóvá lehet tenni. Azért van „rekontra" is: az Eszközök – *Védelem* menüponttal akár egyes lapokat („Lapvédelem"), akár egy teljes munkafüzetet („Füzetvédelem") védetté tehetünk az illetéktelen beavatkozás ellen, jelszóval vagy anélkül. A lapvédelmet programból beállítani az

```
ActiveSheet.Protect Contents:= True, _
DrawingObjects := True, Scenarios := True
```

illetve feloldani

```
ActiveSheet.Unprotect
```

utasítással lehet; az előbbinél külön megszabható, hogy a védelem a cellák tartalmára, a rajzobjektumokra és az esetvizsgálókra (vagy ezek bármilyen kombinációjára) terjedjen ki. Jelszavak programozott kezelésére nincs lehetőség. Mint tudjuk, a *lapvédelem* csak a zárolt objektumokra vonatkozik — a lapok tartományai és objektumai alapértelmezésben zároltak —, ezt a zárolást azonban feloldhatjuk a lapvédelem életbeléptetése előtt a Formátum – *Cellák* (illetve – *Objektum*) menü „Cellák formázása" ablakának „Védelem" lapján. Védett lapon nem lehet az objektumok állapotát megváltoztatni (például sorokat vagy oszlopokat elrejteni vagy felfedni stb.).

A *füzetvédelem* az ablakok geometriai méretére vagy a füzet struktúrájára is vonatkozhat. Ha a füzet struktúráját védjük, nem lehet a lapokat átnevezni, törölni, más füzetbe átmozgatni, vagy új lapokat beszúrni, sem pedig elrejteni, vagy a rejtett lapokat előhívni; továbbá nem lehet az adattáblákat és a kimutatásokat frissíteni. A füzetvédelmet programból beállítani az

```
ActiveWorkbook.Protect Structure:=True,_
  Windows:=False
```

illetve feloldani az

```
ActiveWorkbook.Unprotect
```

utasítással lehet. (A példában csak a füzet szerkezete védett, az ablakok geometriája nem.)

A jelszó nélküli védelem nyilván csak a véletlen rontások ellen véd, hiszen bármikor könnyedén feloldható. A megadott jelszót gondosan jegyezzük meg, mert az Excelből semmilyen módon nem lehet kibányászni; ha elfelejtettük, többé nem tudjuk feloldani a védelmet. Legjobb, ha kiókumulálunk valamilyen algoritmust, amely például a lap vagy a füzet nevére épül, s ennek alapján rekonstruálni tudjuk a jelszót.

Az igazsághoz még az is hozzátartozik, hogy az ilyesfajta védelem csak a konkrét munkafüzetre érvényes: mindig megtehetjük azt, hogy a nem rejtett lapokat átmásoljuk egy másik munkafüzetbe, s ott már semmilyen védelem nem lesz érvényben. Az előbbi kártyás hasonlatnál maradva ez a „szubkontra". Erre már csak egyetlen „licit" van: jelszós szerkezeti védelmű munkafüzetekben a rejtett lap(oka)t csak a jelszó ismeretében lehet felfedni. De ez már kockázatos, mert ha valaki elfelejti a füzetvédelmi jelszót, csak azt teheti, amit az orra előtt elrobogó vonat láttán: bánatosan integethet utána; *a rejtett lapot soha többé nem fogja látni.*

Itt említem meg, hogy a 2. részhez megadott UAFELAD0.XLS állományban is van egy *MODUL* nevű rejtett lap, amelyet a (jelszó nélküli) füzetvédelem feloldása után meg lehet tekinteni. Ezen a lapon vannak a *FÜGGVÉNYEK* munkalap gombjaihoz rendelt rutinok, valamint egy ***Auto_Open*** eljárás, amely egyrészt induláskor kiírja az üzenetet, másrészt szervezőként is működik: az `OnDoubleClick` sajátsággal aktiválja a ***segíts_már*** eseménykezelő rutint. Ennek az lesz a következménye, hogy ha kétszer kattintunk egy függvény nevére, az eseménykezelő rutin automatikusan elindul, és a `Help` metódus segítségével megjeleníti a függvényhez tartozó súgóinformációt. Hogy van-e ilyen (ebben a szűkített listában mindig van), az akkor derül ki, ha a *FÜGGVÉNYEK* munkalapon feloldjuk a (szintén jelszó nélküli) lapvédelmet, és felfedjük a rejtett C oszlopot; itt van az egyes menüpontok kódszáma, amelynek segítségével előkereshető a ...\excel\mainxl.hlp, illetve a ...\excel\macrofun.hlp állományból a megfelelő súgószöveg.

Jóval hatékonyabb az a mentési védelem, amelyet a File - Mentés másként menüpont ablakának EGYEBEK gombjával megnyitott „Mentési beállítások" ablakban lehet beállítani. Ennek 2 szintje van:

— Ha csak írásvédelmi jelszót adunk meg, a megnyitott munkafüzet ugyan módosítható lesz, de az esetleges módosításokat nem lehet visszaírni az eredeti állományba (más néven persze el lehet menteni).

— Ha viszont betekintési jelszót (is) megadunk, az állományt csak a jelszó ismeretében lehet megnyitni; ebben az esetben az átmásolási trükk sem segít.

Van még egy harmadik lehetőség is: a „csak olvasásra ajánlott" figyelmeztetés; ha ezt bekapcsoljuk, az Excel javasolni fogja, hogy a munkafüzetet a felhasználó csak olvasásra nyissa meg; amit vagy megtesz, vagy nem. (A figyelmeztetést rendszerint önmagunknak szánjuk, hogy egy többé-kevésbé végleges vagy kereszthivatkozásokat tartalmazó állományt ne rontsunk el meggondolatlanul.)

Ugyanebben az ablakban jelölhetjük meg azt is, ha állományainkról megnyitáskor biztonsági (backup) másolatot kérünk. Ismerjük Murphy egyik alaptörvényét: ami elromolhat, el is fog. A számítógép is elromolhat, lemezeink is megsérülhetnek, sőt mi több, az Excel is felmondhatja a szolgálatot. Ezért fontos állományainkat mindig biztonsági másolattal nyissuk meg; ha a munka során vagy végén a módosított állomány esetleg megsérül és használhatatlanná válik, legfeljebb a módosítások vesznek el — persze ez sem öröm. A nagyon fontos állományokat pedig mindig két példányban mentsük!

Ezeket a védelmi módszereket programból az

```
ActiveWorkbook.SaveAs Filename := _
  "munkafüzetnév", FileFormat := _
  xlNormal, Password := "főjelszó", _
  WriteResPassword:="írvédőjelszó",
ReadOnlyRecommended := True,
CreateBackup := False
```

parancs lépteti életbe, illetve kapcsolja ki, aszerint, hogy melyik jelszót adjuk meg, illetve adunk a helyén üres ("") füzért, és melyik paraméter értékét hogyan állítjuk be. Külön is felhívom a figyelmet arra, hogy ebben az esetben konkrétan meg kell adni a jelszót, amit persze bárki elolvashat; vagyis a védelemnek csak akkor van értelme, ha az ilyen programot tartalmazó modullapot elrejtjük. (A rejtett lapot viszont füzetvédelmi jelszóval kell védenünk felfedése ellen, amit ha elfelejtünk... jöhet az integetés.)

Munkafüzettípusok

Tekintsük még át, milyen lehetőségeket kínál az Excel munkafüzeteink tárolására.

*„Beépülő" állományok: *.xla és *.xll kiterjesztés*

Az Excel alapállomány nem tartalmaz egy sereg ritkán vagy csak speciális célokra használt eljárást és függvényt; ilyenek például a célértékkereső, a statisztikai elemző stb. rutinok. A rendszerrel szállított ilyen állományok listája az Eszközök – *Makróbeépítő* menüpont választására előugró ablakban tárul szemünk elé; bekapcsolásukhoz egyszerűen ki kell jelölni az állomány neve előtt látható jelölőnégyzetet. (Tartalmukról a súgóban találunk tájékoztatást.) Akinek az alapállomány nem elég, a TALLÓZÓ gomb segítségével kiter-

jedt kutatást folytathat további .xll kiterjesztésű állományok után.

Ez még nem a nóta vége: saját munkafüzeteinkből is készíthetünk beépülő állományokat az Eszközök – *Mentés beépülőként* menüpont választására megjelenő ablakon keresztül. Ez azonban mózesi tevékenység: a kőbevésés modern formája. A beépülőként elmentett állomány .xla kiterjesztést kap, és speciális, csak az Excel által feldolgozható, bináris kóddá alakul, amelynek visszafejtőjét nem fejlesztették ki; vagyis az állomány megváltoztathatatlanul, egyszer s mindenkorra úgy marad. Megjegyzendő, hogy ennek az ablaknak is van EGYEBEK gombja, s ez ugyanahhoz a „Mentési beállítások" ablakhoz vezet, amelyről fentebb esett szó, a védelmi jelszavak megadásával kapcsolatban. Ha .xla kiterjesztésű állományunkat megnyitási jelszóval védjük, abszolút biztosak lehetünk benne, hogy az ebben tárolt titkos információkhoz senki sem férhet hozzá; jelszó nélkül még a nem titkosakhoz sem. Ehhez nem kell bővebb magyarázat, legfeljebb egy jó tanács: mielőtt kőbe véssük, készítsünk a forrásprogramról egy biztonsági másolatot, hátha egyszer mégiscsak meg kellene változtatni...

*Eszköztárállományok: *.xlb kiterjesztés*

Láttuk, hogy lezárásakor az Excel az EXCEL5.XLB állományba menti az aktuális eszköztár-beállításokat, s legközelebb ugyanezzel a beállítással indul. Saját eszköztárak létrehozásáról és tárolásáról a cikk első részében volt szó.

*Normál állományok: *.xls kiterjesztés*

Az eddigi történetek ilyenekről szóltak, nincs további kiegészítés.

*Sablonlapok: *.xlt kiterjesztés*

Űrlapo(ka)t tartalmazó állományaikat célszerű sablonlapként elmenteni a File – *Mentés másként* menüpont ablakának „Fájltípus" mezőjében válasszuk a „sablonlap" típust. Az így elmentett állományt az Excel többé nem nyitja meg közvetlenül, helyette másolatot készít, amelynek nevét úgy képezi, hogy az állomány eredeti nevéhez hozzáilleszt egy sorszámot. Ily módon akárhány példányt megnyithatunk egyszerre, s ezeket tetszőleges más néven menthetjük el. Ezzel kényelmes lehetőséget kapunk arra, hogy űrlapjainkat különbözőképpen töltsük ki; az eredeti sablon épségét az Excel automatikusan megőrzi. (Olyannyira, hogy ha később mégis meg szeretnénk változtatni az állomány kiterjesztését, ezt csak valamilyen kisegítő programmal — például a Norton Editorral — tudjuk megtenni.)

*Állománycsoport: *.xlw kiterjesztés*

Ha egy munkához egyidejűleg több munkafüzetet használunk, nem kell egyenként megnyitnunk és lezárnunk őket. Összes megnyitott munkafüzetünket egyetlen csoportba foglalhatjuk a File - *Munkaterület mentése* menüpont választásával. Az állományokat az Excel természetesen külön-külön menti el, saját nevükön, de RESUME.XLW néven létrehoz egy csoportleíró állományt is. (Ebben az esetben nincs EGYEBEK gomb: az állományokat szükség esetén egyenként kell védenünk. A csoportleíró nevét természetesen megváltoztathatjuk.) Legközelebb a csoportleíró formális megnyitására automatikusan (egyenként) megnyílik a csoporthoz tartozó összes állomány.

Aki már az Excel 4.0 verziójával is dolgozott, tudja, hogy ennek normál állományai ugyancsak .xlw kiterjesztésűek, de minden további nélkül feldolgozhatók az Excel 5.0 rendszerben is. Ezeket megnyitásukkor az Excel automatikusan konvertálja; lezárásukkor viszont felajánlja, hogy .xls változatban menti el. (Amit persze nem kötelező elfogadni; sőt Excel 5.0-s állományokat is menthetünk Excel 4.0 formátumban.)

A felsoroltakon kívül a „Mentés másként" ablak „Fájltípus" mezőjében még további 21-féle típus közül választhatunk, amelyekben elmenthetjük munkánkat (vagy egy részét), ha a feldolgozást más rendszerekkel akarjuk folytatni.

Biztató befejezés

A cikksorozat végéhez érvén bánatosan kell megállapítanom, hogy a rendelkezésemre álló viszonylag nagy terjedelem ellenére sok kérdésről nem, vagy csak érintőlegesen esett szó; főleg a lehetőségek irdatlan nagy száma miatt, de azért is, mert mindegyiket magam sem tudtam eleddig kipróbálni. Mégis bízom abban, hogy sikerült megmutatnom a módszert, hogyan oldanék meg egy problémát az Excel és a VBA nyelv segítségével: ha tudom, milyen metódusokat, eljárásokat, függvényeket, utasításokat használjak, megkeresem használatuk szabályait (esetenként egyes összetettebb lépések kialakításához felhasználom a makrógenerátor segítségét); ha viszont nincs konkrét elképzelésem, hasonló megoldásokat keresek a súgóban: minden tételhez van „lásd még" („see also") és mintapélda („example"). Itt említem meg azt is, hogy a modullapokról az F2 billentyűvel vagy a Nézet – *Objektumtallózó* menüponttal megnyitott „Objektumtallózó" ablakban nemcsak saját nyitott munkafüzeteink objektumainak listája látható, hanem a „Rutinkönyvtárak/Munkafüzetek" mezőben választhatjuk a VBA vagy az Excel könyvtárait is, és ekkor megkapjuk az összes lehetséges VBA, illetve Excel objektum összes lehetséges metódusának és sajátságának listáját. Ha ezek közül egyet kijelölünk, a ? ikonra kattintva megnyílik a hozzá tartozó súgó, amelyben mindig találunk mintapéldákat is.

Tanulságként hadd említsem meg a lopódarázs példáját. Ez a rovar úgy gondoskodik az utódairól, hogy egy földbe vájt üregben rakja le a petéit, és melléjük temet egy tücsköt vagy pókot, esetleg hernyót, amelyet előzőleg mérgével megbénít. Amikor a lárvák kikelnek, vígan felfalják a még élő konzervet, s mire elfogy, annyira felnőnek, hogy már maguk is képesek megszerezni táplálékukat. A megoldás bámulatra méltóan zseniálisnak látszik; de néhány kibírhatatlan rovartanász beleszólt a folyamatba. Amikor ugyanis a fészek kész, a lopódarázs odakészíti a megbénított rovart a bejárathoz, bemegy elvégezni a peterakás előtti végső simításokat, majd visszatér, hogy becipelje az élelmet, és befejezze a művét. Ezalatt azonban az említett rovartanászok egyike egy fűszállal elpiszkálta a konzervet a fészek nyílásától, amivel megoldhatatlan probléma elé állította az utódaiért fáradozó állatot: mikor ez kijőve nem találta ott a korábban megszerzett zsákmányt, szemmel láthatóan megzavarodott; lehet, hogy meg is döbbent, de ez nem biztos. Elkezdett keresgélni, s ha megtalálta újból odacipelte a fészek elé a béna rovart; rossz esetben keresett másikat. Ezután *visszament a fészekbe elvégezni a peterakás előtti végső simításokat*. Ha ezenközben az említett rosszindulatú egyén ismét elkotorta a félkész élelmet a fészek elől, a fenti ciklus akárhányszor megismétlődött, amíg a tudomány derék művelője meg nem unta a játékot. A lopódarázs nem volt képes felismerni, hogy az előkészítő műveleteket már elvégezte: ha hiányzott az odakészített élelem, mindent kezdett elölről.

Úgy gondolom, hogy a világ leggyengébb programozója sem esnék hasonló hibába; abban pedig biztos vagyok, hogy annál, aki végigrágta magát ezen a sorozaton, ilyen eset fel sem merülhet; hiszen nem ösztönszerűen cselekszik, hanem előre megtervezi a programját; s ha el is akad, mindig pontosan tudja, milyen résztevékenységeken jutott már túl. Legalábbis mostantól fogva — remélem.

Álló Géza

E SZÁMUNK HIRDETŐI

A Mikrobazár rovatban a nem kereskedelmi célú egyéni hirdetések közlése ingyenes.

A kereskedelmi célú apróhirdetések tarifája gépelt soronként (azaz 60 karakterenként) 300 forint.

A terjedelem alapján így kiszámított összeget kérjük átutalni az Új Alaplap Kiadói Kft számlájára (OTP, 11706016–20788599), vagy feladni postai utalványon a kiadó címére (1539 Budapest, Pf. 571), és feltüntetni, hogy „Új Alaplap, apróhirdetés". A befizetést igazoló szelvény másolatát — a hirdetési szöveggel együtt — a szerkesztőséghez (a kiadóéval azonos címre) küldjék el.

Szerzői jogokat sértő szoftverhirdetéseket nem közlünk le.

Bármilyen típusú szöveg fordítását vállalom angolról magyarra, magyarról angol nyelvre, illetve vállalom kiadványok látványtervezését, szerkesztését is. Cím: Lachner Zoltán, 1195 Budapest XIX., Jáhn Ferenc u. 14/a. Telefon: 357-0308.

OBJECTS 2.0 — objektumorientált programozás CLIPPER-ben. Tájékoztató kérhető az alábbi címen: Szűcs János, 4400 Nyíregyháza, Vasvári Pál u. 37. Tel.: (42) 437-331 vagy 465-666/1382-es m.

Adatmentés CD-re, streamerre; winchesterről, floppyról. Ugyanitt beszerzési tanácsadást, hálózattervezést és programkészítést is vállalok. Cím: Kovács Lajos, 1031 Budapest III., Vízimolnár u. 10. IV/33.

Alaplapcsere, memória-, winchester- és floppybővítés a helyszínen. MegaSoft. Telefon: 295-5085.

Stúdiómban megbízhatóan, ellenőrzötten lefordítom angol, német, francia és magyar nyelvről/nyelvre műszaki és közgazdasági folyóiratok cikkeit, hardver- és szoftverleírásait. Áfás számlát állítok ki. Cím: Szász György, 1035 Budapest III., Kórház u. 25. Tel.: 368-4874.

Akarod, hogy ingyen tiéd legyen az évszázad viccgyűjteménye? Nos, ha igen, akkor írj a kgb@server.gaboraron-misk.sulinet.hu címre, vagy keress telefonon (20)970-3438!

Keresek 10"-os VGA-LCD kijelzőt notebook gépemhez. Típusa: Extensa 450. Az összetört LCD típusjele: Sharp-LM64C38P. Cím: Májer István, 8360 Keszthely, Rákóczi u. 48. Tel./Fax: (83)318-345.

Elcserélem a PC World magazin 1997/1, 2, 3, 4, 8, 9 számait, CD-ROM melléklettel együtt 3 darab írható CD-lemezért (CD-R) vagy eladom 1000 Ft + postaköltség áron. Cím: Ag Júlia, P.O.Box 155, 92901 Dunajská Streda, Slovensko.

Új digitális fényképezőgép, Panasonic PV-DC1000, eladó 100 ezer forintért. Jellemzői: 2 MB beépített memória, 46 mm color LCD, 350000 PS CCD, 640x480 képfelbontás, superfine–fine–normal üzemmód, 16-34 kép, JPEG formátum, fix fókusz, normál és macro. Külméret: 9x6x3 cm. Digitális és video jelkimenet. Telefon: 383-0635.

Vennék Cyberdrive Infra 18x CD-ROM meghajtót, lehetőleg távkapcsolóval. Intelligens Assembly compiler is érdekel. Érdeklődni lehet: (30)933-7939.

Eladom 486 DX 100-as gépemet, nagy toronyban, 40 MB winchesterrel, 16 MB RAM-mal, EGA videokártyával, EGA monitorral. Irányár: 25 000 forint. Érdeklődni: (20)979-0202.

Pascal forráskódokat és shareware, freeware programokat cserélek. Mátrai Ferenc, 3033 Rózsaszentmárton, Kossuth u. 37. Tel.: (37)384-459.

Adatmentés CD-re. Ugyanitt shareware, freeware listából válogatási lehetőség. Kérésre e-mailen listát küldök: abonyim @mail.matav.hu

Gyűjteményembe keresem megvételere a Kék lagúna, The Nightmare before Christmas és a Star Kid című filmeket, kazettán vagy video-CD-n. Keresek továbbá valakit, aki tud videokazettáról **CD-re íratni.** Hasuly Balázs, telefon: 340-1419.

„Barkács" utasítások

Az animáció gyorsítása

A „komondor" gépek jól fel voltak készítve a játékokra. Könnyűszerrel lehetett velük kalandjátékokat, látványos ütközéseket, lövöldözéseket produkálni. A könnyű programozás titka, hogy az ún. „manócskák", „szellemek" (sprite-ok) megjelenítését hardverrel támogatta a C64 videochipje, a VIC-II. A képernyő tetszőleges helyén létre lehetett hozni ilyeneket egy-egy 24x21-es pixelméretű mezőben. A chip hipp-hopp beállította a sprite regisztereit, de megszakítást generált, ha a sprite a háttérrel vagy egy másik sprite-tal ütközött. Ismerjük el: a C64 korát jóval megelőzve vált alkalmassá animációk készítésére.

A Simon's Basic — emlékszik még rá valaki? — a VIC II. videochipre alapozta számos kényelmes sprite-utasítását, megkímélve a programozókat a fáradságos számolgatásoktól. Hála a hardvertámogatásnak, a sprite-utasítások villámgyorsan működtek, legalábbis a Basic programokhoz képest. Minden adottság megvolt tehát ahhoz, hogy könnyűszerrel biztosítani lehessen a mozgás illúzióját.

Szoftveres manócskák

Az animáció ma is kulcskérdése a játékkészítésnek. A mai chipek hardverrel nem támogatják az efféle manócskák létrehozásához és gyors mozgatásához szükséges műveleteket — ha tehát ilyesmire vágyunk, magunknak kell létrehoznunk szoftveres szerszámosládát az ilyen jellegű hatások megvalósításához. Ne felejtsük el, hogy a folyamatos mozgás érzékeléséhez (tehát szemünk becsapásához) a monitornak másodpercenként legalább 25-ször kell frissítenie a képet, hogy sem darabosnak, sem villogónak ne érezzük a változásokat. Közben pedig két kép kialakítása között esetleg több százezer elemi művelet elvégzésére lehet szükségünk!

A program végrehajtásának egy-egy kritikus pontján bizony kevésnek bizonyulhat gépünk sebessége. No nem azért, mintha eleve lassúak lennének a mai processzorok, hanem a rengeteg kitérő, tili-toli és biztonsági ellenőrzés miatt. Az átlagos sebességcsökkenés, amit egyszerűen a magasszintű nyelvek használata okoz, megközelíti, néhol meg is haladja az egy nagyságrendet. A kép kialakításának vagy megjelenítésének felgyorsítására tehát vagy különleges megoldásokat kell alkalmaznunk, vagy közelebb kell kerülnünk a közvetlen gépi programozáshoz. A C nyelv jó segítséget nyújthat ebben, de csak annak, aki szakavatott módon használja. Egyszerűbb és fájdalommentesebb megoldás, ha a számítástechnika Hamupipőkéjéhez, a már-már elfeledett Assembly nyelvhez fordulunk segítségül.

Barkácsnyelvű Turbo Pascal betétek

Kellemes újítása volt a Turbo Pascalnak, hogy lehetővé tette a „barkácsnyelvű" (Assembly) kódrészletek közvetlen beírását a Pascal forrásnyelvű programokba. Ezek fordításáról és a program többi részével való egybeolvasztásáról a 6.0 változat óta automatikusan gondoskodik egy beépített assembler, a BASM (Buil-in ASM), itt-ott még kódoptimalizálást is végez. Komoly könnyebbséget jelent, hogy az Assembly kódrészletekben használhatjuk a Pascal változókat, konstansokat, címkéket, de még a függvényeket és eljárásokat is. Hivatkozhatunk az aktuális kódszegmens és adatszegmens értékére, sőt ha stringekről van szó, akkor a függvény visszatérési értékének címére is.

A BASM használata tehát nagyon kényelmes. Egyetlen komoly hátránya, hogy utasításkészlete szűkebb, mint elvárhatnánk tőle. A Pascal beépített utasításokként csak a 80286 processzor utasításait engedi meg, kimaradnak tehát a körből a 386-os és 486-os processzorok 32 bites utasításai. (Ez a kellemetlen korlátozás természetesen megkerülhető, ha a hagyományos utat járjuk, és a megszokott módon object fájlokként szerkesztjük bele a programba az ilyen Assembly betéteket. De így hol marad a kényelem?)

Van egy másik rigolyája is a Pascalnak, amit el kell viselnünk: minden programra (még a védett módúakra is) ráerőlteti a maga sajátos memóriakezelését. Ezért például bajunk támadhat a Pascal programban a lokális változók elérésével, ha a beültetett Assembly kódrészletben orvul megváltoztatjuk a BP regiszter tartalmát — a változók és a paraméterek hívásához ugyanis a Pascal mindig ezt a regisztert használja. Ugyanígy bajunk lehet a globális változók elérésével is, ha óvatlanul bánunk a DS regiszterrel.

Néhol viszont éppen az Assembly kódrészletek segíthetnek a Pascal kötöttségeinek feloldásában: ez a helyzet például a memóriaallokáló utasításokkal kapcsolatban. Mint tudjuk, dinamikus memóriaterületként a Pascal programok az ún. „heap"-et használják. Ennek korrekt kezelése alapvető fontosságú például az objektumorientált programozásban, hiszen a dinamikus memóriában jönnek létre és szűnnek meg a dinamikus változók. Normális esetben a Pascal program heapje „rátenyerel" az összes rendelkezésre álló memóriára (védett módban meg a teljes fizikai memóriára), a programoknak úgy kell „kikönyörögniük" a heaptől a kellő méretű memóriát, dinamikus memóriaallokáló utasításokkal. A probléma ott kezdődik, hogy a Pascal eszközeivel nem tudunk egyszerre sem lefoglalni, sem felszabadítani 64 KB-nál nagyobb összefüggő területet. Ilyenkor vehetjük hasznát a barkácsnyelvnek, amikor a Pascal tudománya már éppen elfogy. (A korrekt memóriakezelés megvalósítása persze nem mindig könnyű, védett módban különösen komoly szakmai ismereteket feltételez.)

Kisebb programoknál nyilván ritkábban merülnek fel rázós problémák. Mi is maradhatunk egyelőre a magunk kis szemétdombján, ne firtassuk a részleteket, elégedjünk meg annyival, hogy újabb elemekkel bővült Pascal programozói eszköztárunk. Eszünkbe se jusson lemondani a kényelmes és áttekinthető Pascal programozásról, hiszen a beépülő barkácsnyelvű kódrészek alapvetően nem változtatják meg, csak ki-

egészítik programozói környezetünket. Máskor nem is érdemes átváltanunk barkácsnyelvre, csak akkor és ott, ahol döntő kérdés a sebesség. Azt is megnyugvással konstatálhatjuk, hogy az Assembly betétektől még az egész programrész „külsőleg" megmarad Pascal nyelvűnek. Készíthetünk tehát belőlük könnyen kezelhető könyvtári egységeket (TPU unitokat), amelyeket csak olyankor építtetünk bele programjainkba, ha szükség van rájuk.

Spékelt TP a gyakorlatban

Hardverproblémák megoldásához a ComputerBooks már több értékes kiadványban népszerűsítette ezt a módszert: László József külön könyvekben mutatta be a perifériák, hangkártyák és VGA kártyák programozását Pascal és Pascalba oltott Assembly nyelven. Most ez a „de facto sorozat" egy újabb darabbal gazdagodott: Varga Márton leírja, hogyan hozhatók létre komplett kis rendszerek speciális célokra, olyanokra, mint például a játékkészítés. A most publikált kötetben két menüvezérelt editor is található, az egyik „szoftver sprite-ok", az ún. BOB-ok szerkesztésére szolgál, a másik, a MAP-editor, a háttér kialakítására. A fő attrakció azonban a GAME unit. Ez egy olyan programkönyvtár, amely majdnem háromtucatnyi eljárással és függvénnyel segíti Turbo Pascal nyelven a játékkészítést, ezen belül (figyelem!) az objektumként definiált BOB-ok kezelését is.

Mit tudnak ezek a BOB-os megjelenítők? Íme, egy rövid felsorolás:

— Egyszerre több BOB és több háttér is használható a képernyőn.

— A háttér a grafikus objektumoktól teljesen függetlenül kezelhető, tetszőlegesen változtatható (például görgethető).

— A BOB-ok helyzete, mérete vagy animációs fázisa tetszőleges lehet.

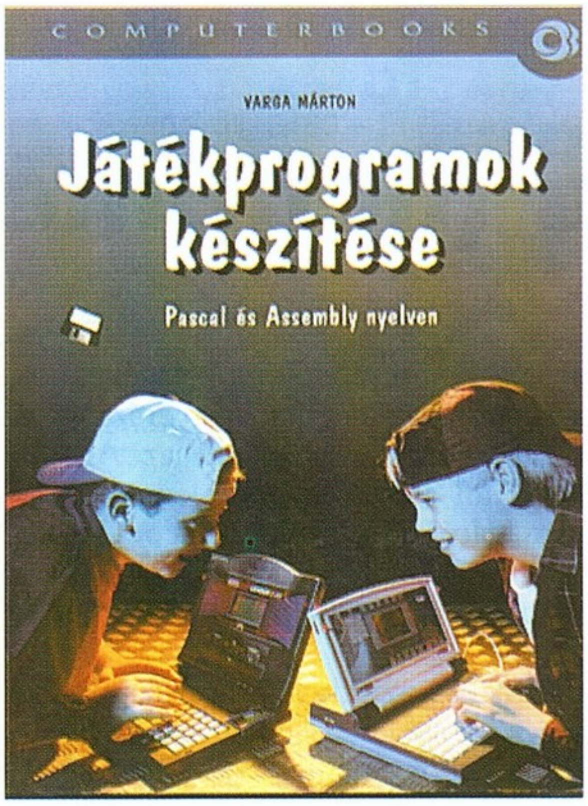
— Bizonyos részein átlátszhat a háttér vagy egy másik BOB takaratlan része.

— Egymásra helyezett BOB-ok között változtatható a prioritás, hogy melyikük kerüljön felülre.

— A villogás és a darabosság minimálisra csökkenthető.

— Megoldható a BOB-ok részleges megjelenítése (például a képernyő szélén).

— Könnyen kezelhető a BOB–háttér és a BOB–BOB ütközés, sőt egy harmadik típusú ütközés is, ahol adott méretű térképegységekből (az ún. BOX-okból) megnövelt játékteret alakítunk ki, és az ilyen BOX példányokkal való ütközést figyeljük a megnövelt



játéktérben. A BOX-okból felépített játéktérnek tehát nem kell elférnie a képernyőn, elég, ha egyszerre annak csak egy részletét is meg tudjuk jeleníteni.

A könyvben feladatok sorozatán keresztül vezet el a szerző az igazán igényes kihívásokig, a legegyszerűbb megjelenítési problémából kiindulva. A feladatok általában TP-be ágyazott barkácsnyelvű programokkal oldhatók meg, de nem kell megijedni tőlük, minden lépés bőségesen el van látva magyarázatokkal. Ráadásul a könyv floppymellékletén a forrásnyelvű változatok mellett a gépi rutinok is megtalálhatók. Aki tehát megelégszik a kész TP rutinok használatával, annak az Assembly nyelvben sem kell elmélyednie, bőségesen elég, ha átfutja az Assembly betétek funkciójának leírását. A lemezmellékleten egyébként szemléltetésként két szándékoltan egyszerű mintaprogram is található, közvetlenül futtatható formában, hogy mindjárt kóstolót is kapjunk a programkönyvtár felhasználási lehetőségeiből.

Érdemes megjegyezni, hogy a programok védett módban nem futtathatók, mivel változók vannak egyes kódszegmensekben. Hogy miért vannak? A magyarázat egyszerű: a futási idő megrövidítésének kényszere kívánja ezt a megoldást. Ugyanis viszonylag sok időbe telne, ha a megjelenítési cikluson belül mindig vissza kellene tölteni az eredeti adatszegmens regisztert.

Ambiciózus programozóknak

Igényesebb programozók számára a mintaprogramok és az editorok futtatásából is látható, hogy két út kínálkozik a bemutatott rendszer továbbfejlesztésére. Az egyik út a háromdimenziós objektumok ábrázolásával kapcsolatos feladatok irányába vezet, a másik a felbontás finomításának irányába. A szerző ugyanis nem akarta túlbonyolítani a feladatot. Az alapprobléma megoldása azáltal is tisztábban áll előttünk, hogy a szerző megelégedett a kétdimenziós játékok grafikai problémáinak elemzésével (ez is nyújt éppen elég meggondolnivalót), és, valljuk be, az is érthető, hogy a finomítási problémák dzsungelébe nem kívánt belebonyolódni. A megjelenítési lehetőségek sokféleségének hosszadalmas tárgyalása helyett mindenütt csak a könnyen kezelhető MCGA üzemmódot használja. Ez standard módon minden VGA kártyán megtalálható, és bár kétségtelenül elég durva, színekben mégis gazdag megjelenítést biztosít.

Ami a háromdimenziós megjelenítést illeti, ez önmagában is hatalmas téma. Ambiciózus olvasóink hasznos segítséget kaphatnak a továbblépéshez Agárdi Gábor „haladó barkácsolók" számára készített könyvéből (ismertetését lásd alább) vagy Füzi János könyveiből (3D grafika és animáció IBM PC-n, illetve Interaktív grafika). Ez utóbbi könyvekről több ízben adtunk részletes ismertetést (1996/5., 1997/4., 1998/1. szám) — érdemes utánanézni a kínált lehetőségeknek és az ajánlott megoldási módoknak.

Más jellegű meggondolások vezették a szerzőt, amikor lemondott a felbontás finomítási lehetőségeinek tárgyalásáról. Tagadhatatlan, hogy az MCGA üzemmód a maga 320 x 200-as felbontásával bizony elég szerény teljesítmény. Hogy mégis közkedveltté vált (és nem csak a játékprogramírók körében), az egyszerű gyakorlati okokkal magyarázható. (1) A szabványos VGA grafikus módok közül egyedül ez tud 256 színt egyszerre megjeleníteni. (2) Kezelése elég egyszerű, mert nem vesz igénybe 64 KB-nál több memóriát. (3) Minden VGA rendszer érti és egységesen kezeli.

Nyilvánvalóan jobb teljesítmény lenne elérhető SVGA kártyákkal, de egyrészt nincs mindenkinek ilyen kártyája, másrészt problémák adódnak a különböző gyártmányú kártyák kezelésének sokféleségéből. Mint tudjuk, elkéstek ezeknek a szabványosításával, így azután az egyes gyártók ahány grafikus kártya, annyiféle módon határozták meg az ellátandó funkciók körét és megvalósítási módját. Csak néhány az ismertebb rendszerek közül: Paradise, NCR, Trident, Everex, ATI, Video7, Chips & Tech, Genoa GVGA, és a sor ezzel még távolról sem ért véget.

Vargha Dénes

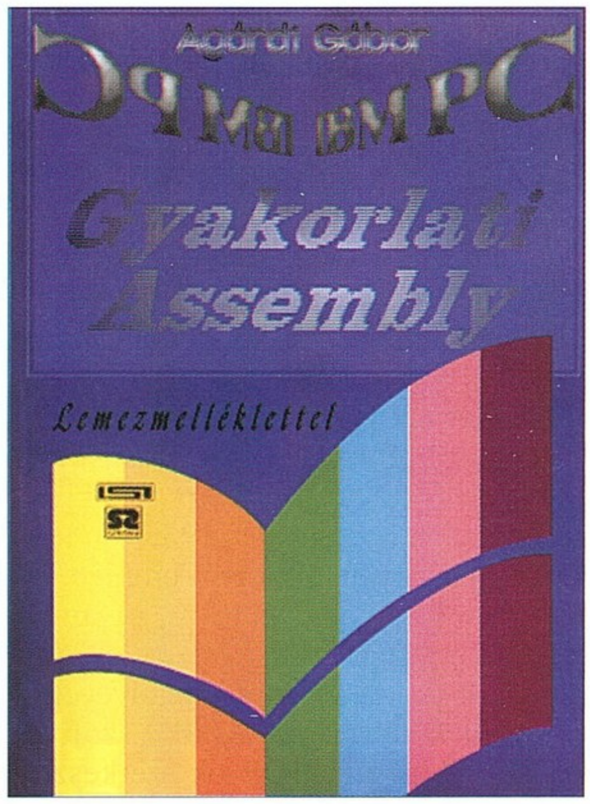
Assembly-kínálat

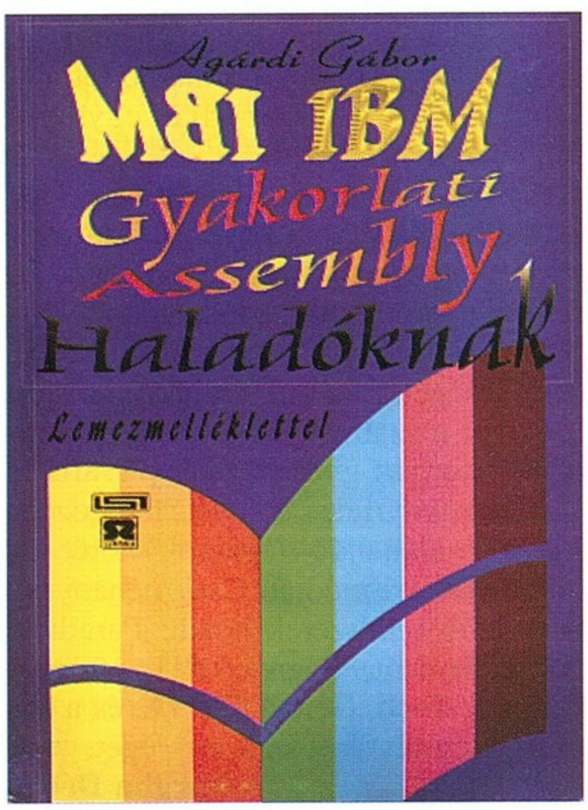
Tartalom és (nem méltó) forma

Nem különösebben gazdag a könyvkínálat az Assembly nyelv ismertetése terén. Ma már unikumnak számít Gidófalvi Zoltán műve (Az IBM PC programozása Assembly nyelven, Novotrade, 1988), és ha nem is avult el, az újabb dolgokat természetesen nem tartalmazza. Igen alapos munka dr. Kovács Magda kétkötetes ismertetése a 32 bitre bővített mikroprocesszorokról (32 bites mikroprocesszorok, 80386/80486 I-II, LSI Oktatóközpont, 1991, 1994). Ez tartalmazza a teljes utasításkészlet részletes bemutatását, beleértve a védett módú utasításokat is — de a barkácsnyelvű programozás elsajátításához nem biztos, hogy ez kell. Ami ebben a könyvben található, az részben sok, részben kevés.

Kellemes középutat jelentenek Agárdi Gábor könyvei az elmélet és a gyakorlat arányának meghatározásában. Szimpatikus a problémaorientált megközelítés és a feladatok egymásra épülése. Nem célja a szerzőnek az Assembly utasításkészlet teljes repertoárjának bemutatása, a bemutatott példák nyomán azonban az alapokat valóban jól el lehet sajátítani.

A kezdők számára szóló kötetben a teljes anyagnak a felét az Assembly grafika ismertetése tölti ki. Aki netán kifogásolta, hogy az előző könyv szerzője a játékprogramok készítésének kapcsán miért nem ment be a különböző megjelenítési módok rengetegébe, nos, az itt megtalálhatja a magyarázatot. A teljesség nem várható el az Agárdi-könyvtől sem, de a probléma bonyolultságáról eléggé világos képet kaphatunk a CGA, EGA, VGA, SVGA különböző üzemmódjainak ismertetése alapján.





Érdekes megfigyelni, hogy mindkét szerző lényegében ugyanazt a tárolási formátumot ajánlja az olvasóknak. Agárdi a DeLux Paint rajzolóprogram képformátumaként vezeti be, és hol DP, hol LBM formátumként hivatkozik rá, Varga egyértelműen LBM formátumról beszél. A BMP formátumtól mindketten óva intenek nagy helyfoglalása miatt, a GIF és a JPG formátum ellen pedig az szól, hogy az így tárolt képek nehezen és viszonylag lassan jeleníthetők meg. Ráadásul a GIF fájlokban alkalmazott LZW kódolás jogvédett, ezért nem árt, ha nagy ívben elkerüljük. Az LBM formátmának nagyon egyszerű a tömörítési módja, az esetek többségében mégis jó sűrítési arányt biztosít, a szélsőséges ellenpéldák ellenére. Módszerének az a lényege, hogy az egymás utáni azonos színeket vonja össze. Nyilvánvaló tehát, hogy a várt sűrítés helyett az eredeti képnél is nagyobb terjedelmű fájlt kapunk, ha történetesen minden képpont más színű, de a képek általában nem ilyenek.

Agárdinak a haladók számára írott könyve erősen épít előző kötetére. Értékes többlete az animációkról és a háromdimenziós alakzatok kezeléséről szóló rész. Bemutatja, hogyan lehet Assembly nyelven zárt alakzatokat kifesteni, ellipszist rajzolni, hangeffektusokat kelteni. Csupa hasznos dolog, ha nem is különösebben izgalmas. Érdekes viszont az a rész, ahol felvillantja, hogy a vírusírók milyen módon igyekeznek akadályozni a kód visszafejtését.

Agárdi könyvét olvasva nem lehet szó nélkül elmenni a szerkesztési figyelmetlenségeken, mert azok bizonyos elfogadható mértéket meghaladó arányban bosszanthatják azokat, akik még egyáltalán figyelnek ilyesmire. Ide tartozik egyrészt olyasmi, hogy az első könyvnek csak a tartalomjegyzékében szerepelnek az egyes részek alcímei, a szövegből (technikai okokból?) kimaradtak. Még inkább azonban a magyar helyesírás központozási szabályainak enyhén szólva lezser kezelése, vagy olyan szóalakok előfordulása, mint „válltoztatja", „kiválló", „álltalunk", vagy az installáló lemezen az egyik képernyőre írandó szövegben a „befelyezi". Egy könyv tartalmilag is veszít hitelességéből, ha abban bizonyos „egyéb" normákat túl gyakran megsértenek.

Vargha Dénes

Varga Márton:
Játékprogramok készítése Pascal és Assembly nyelven
ComputerBooks, 1998
196 oldal, 1456 Ft
(Lemezmelléklettel)

Agárdi Gábor:
IBM PC Gyakorlati Assembly
LSI Oktatóközpont
212 oldal, ár megjelölése nélkül
(Lemezmelléklettel)

Agárdi Gábor:
Gyakorlati Assembly haladóknak
LSI Oktatóközpont
204 oldal, 1076 Ft
(Lemezmelléklettel)

ELŐFIZETÉS

Az 1999/......... számtól kezdődően előfizetem

az Új Alaplap című CD-mellékletes havi számítástechnikai folyóiratot

...... példányban ❑ 1 évre, ❑ 1/2 évre.
Az éves előfizetési díj: 6480,– Ft (Ez az összeg az áfát is tartalmazza.)

❑ **Számlát** kérek (banki átutalással fizetek). ❑ **Befizetési csekket** kérek.

Név: ....

(Cég:) ....

Cím: ....

Irányítószám, helység: ....

Dátum:

/aláírás/

APRÓHIRDETÉS

Kérem, hogy az Új Alaplap következő számának Mikrobazár rovatában az alábbi szövegű apróhirdetést jelentessék meg. (A túloldalon ismertetett feltételeket tudomásul veszem.)

....

....

....

....

....

....

....

....

(Maximális terjedelem 300 betű.)

INFORMÁCIÓKÉRÉS

Az Új Alaplap mostani számában megjelent hirdetések közül az általam itt megjelölt kódszámúakhoz részletesebb információt kérek a hirdető cégektől.

01	02	03	04	05	06	07	08	09	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81	82	83	84
85	86	87	88	89	90	91	92	93	94	95	96

Új Alaplap, 1999. februári szám. Beküldési határidő: 1999. február 28.

Belföldön díjmentesen is feladható

ÚJ ALAPLAP

VI., Dózsa György út 84/b
Postafiók 571
1539 Budapest

Feladáskor kérjük bérmentesíteni!

FELADÓ:

Név:...

Cím:...

Helység: ....

Irányítószám:...

Telefon: ....

☐ A hirdetés egyéni és egyedi jellegű, ezért kérem ingyenes megjelentetését. Kijelentem, hogy annak tartalma nem sérti senki szerzői jogát.

☐ A hirdetés kereskedelmi célt szolgál. Mellékelem a soronként (60 karakterenként) 300 forintnak megfelelő összeg átutalásáról az igazoló szelvény másolatát. A címzett: Új Alaplap, 1539 Budapest, Pf. 571, illetve átutalásnál az OTP 11706016-20788599 számlaszámra.

- - - - - - - - - - - - - - - - - - -
/aláírás/

Bélyeg helye

ÚJ ALAPLAP

VI., Dózsa György út 84/b
Postafiók 571
1539 Budapest

FELADÓ:

Feladáskor kérjük bérmentesíteni!

a) EGYÉNI

Név: ....

Cím: ....

Helység, ir.sz.: ....

b) CÉGES

Név: ....

Cég: ....

Cím: ....

Helység, ir.sz.: ....

Telefon: ....

- - - - - - - - - - - - - - - - - - -
/aláírás/

Bélyeg helye

ÚJ ALAPLAP

VI., Dózsa György út 84/b
Postafiók 571
1539 Budapest

HP Vevőszolgálat: 343-0310 http://www.hp.hu/periferiak/





MEGTÉVESZTŐEN VALÓSÁGHŰ.



Ezt a rendkívül életszagú képet a világ jelenlegi legfejlettebb tintasugaras nyomtatási eljárásával állították elő. Vagyis a HP PhotoREt II színrétegező technológiájával. Ez az az új technikai megoldás, amely lehetővé teszi nyomtatóink számára, hogy akár 16 miniatűr tintacseppet keverjenek és juttassanak el minden apró képpontba, s ennek köszönhetően a színek összes elképzelhető árnyalata rendelkezésre álljon a lenyűgöző minőségű, fotórealisztikus képek készítéséhez – még hétköznapi, irodai papírra nyomtatva is. És még ennél is bámulatosabbá teszi a dolgot az a tény, hogy mindez olyan sebességgel működik, amellyel más asztali printerek csak fekete-fehér nyomtatásra képesek. Egy szó mint száz, a HP PhotoREt II-nek más a nyomába sem ér. Kivéve persze magát a valóságot.

HP DESKJET NYOMTATÓK
HP PhotoREt II technológiával

HEWLETT PACKARD

Expanding Possibilities